



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV MIKROELEKTRONIKY

DEPARTMENT OF MICROELECTRONICS

**SYSTÉM PRE SLEDOVANIE POHYBUJÚCICH SA
OBJEKTŮV**

MOVING OBJECTS MONITORING SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jakub Orolin

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Jiří Háze, Ph.D.

BRNO 2019

Diplomová práce

magisterský navazující studijní obor **Mikroelektronika**
Ústav mikroelektroniky

Student: Bc. Jakub Orolin

ID: 173717

Ročník: 2

Akademický rok: 2018/19

NÁZEV TÉMATU:

Systém pre sledovanie pohybujúcich sa objektov

POKYNY PRO VYPRACOVÁNÍ:

Vytvorte autonómny zameriavací systém pohybujúcich sa objektov určený pre kamery alebo mobilné telefóny. Úlohou systému je smerovanie kamery systému na pohybujúce sa objekty ako je lietadlo, lyžiar na svahu apod. Vyberte vhodnú platformu a navrhnete možné riešenia implementácie. Systém realizujte a overte jeho funkčnosť v reálnom prostredí.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

Termín zadání: 4.2.2019

Termín odevzdání: 21.5.2019

Vedoucí práce: doc. Ing. Jiří Háze, Ph.D.

Konzultant:

doc. Ing. Lukáš Fujcik, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Predložená diplomová práca je zameraná na návrh a realizáciu systému schopného sledovať pohybujúce sa objekty. Výstupom práce je prototypové zariadenie, ktoré je fyzicky umiestnené medzi kamerou a statív a otestované v reálnych podmienkach. Úlohou tohto systému je automatické natáčanie kamery na pohybujúci sa objekt.

ABSTRACT

The presented thesis deals with the design of a system capable of tracking the moving objects. The output of the thesis is the prototype layout of the device. Facility will be physically placed between the camera and the tripod in the dissertation and tested in real conditions. The role of this system is to automatically rolling the camera up the selected moving object.

KLÚČOVÉ SLOVÁ

Kamera, Statív, FPGA, Sledovanie objektov, Spracovanie video obrazu, Krokový motor, Servomotor

KEYWORDS

Camera, Stand, FPGA, Object tracking, Video processing, Stepper motor, Servomotor

OROLIN, Jakub. *Systém pre sledovanie pohybujúcich sa objektov*. Brno, 2019. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/119422>. Diplomová práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav mikroelektroniky. 2019. 67 s. Vedúci práce doc. Ing. Jiří Háze, Ph.D.

Prehlásenie autora o pôvode diela

Prehlasujem, že svoju diplomovú prácu na tému Systém pre sledovanie pohybujúcich sa objektov som vypracoval samostatne pod vedením vedúceho diplomovej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušenia ustanovení § 11 a nasledujúceho autorského zákona č. 121/2000 Sb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka č. 40/2009 Sb.

V Brne dňa 21. Mája 2019

.....
podpis autora

Pod'akovanie

Ďakujem vedúcemu diplomovej práce Doc. Ing. Jiřímu Házemu, Ph.D za cennú metodickú, pedagogickú a odbornú pomoc. Ďakujem i svojej sestre Janke za štylistickú korektúru textu.

V Brne dňa 21. Mája 2019

.....
podpis autora

Experimentálna časť tejto diplomovej práce bola realizovaná na výskumnej
infraštruktúre vybudovanej v rámci projektu CZ.1.05/2.1.00/03.0072

Centrum senzorických, informačných a komunikačných systémov (SIX)
operačného programu Výskum a vývoj pre inovácie.

OBSAH

Úvod do problematiky	1
1 Mechanický koncept SSPO.....	2
2 Prehľad komerčne dostupných produktov	3
3 Princíp sledovania pohybujúcich sa objektov z videosekvencie.....	4
4 Výpočet maximálnej rýchlosti objektu pre naše riešenie.....	6
5 Definícia obrazových formátov.....	8
6 Princíp extrakcie pohybujúcich sa objektov z video sekvencie	10
7 Odstránenie šumu z rozdielového snímku	12
7.1 Zdroje šumu na vstupných snímkach.....	12
7.2 Vplyv vstupného šumu na rozdielové snímky	12
8 Výpočet stredu objektu z množiny bodov.....	15
9 Praktická realizácia SSPO	17
9.1 Návrh mechanicko-motorickej časti	17
9.2 Výber testovacej kamery	18
9.3 Výber vhodnej realizačnej platformy a jej parametre.....	19
9.4 Koncept základnej dosky plošných spojov pre RJ.....	20
9.5 Koncept RJ v programovateľnom hradlovom poli FPGA	21
9.5.1 DVP dekodér	22
9.5.2 Prevodník RGB na Y	28
9.6 Návrh testovacej platformy algoritmov sledovania	30
9.7 Distribúcia napájania s ochranou batérie	34
9.8 Implementácia detekcie pohybujúceho sa objektu v .NET aplikácii.....	37
9.9 Filtrácia malých objektov v binárnom obraze	38
9.9.1 Verifikácia entity filtration_unit.vhd.....	42
9.10 Generátor PWM impulzov pre riadenie servomotorov.....	44
9.11 Vysokorýchlostný verifikačný kanál FT245 s generátorom správy	45
9.12 Implementácia jednotky pre separáciu statického pozadia.....	50
Záver	53
POUŽITÁ LITERATÚRA	55
Zoznam symbolov veličín a skratiek	57

Zoznam obrázkov	59
Zoznam tabuliek	60
A Schematický návrh SSPO	61
A.1 Zapojenie konektorov s distribúciou napájania	61
A.2 Zapojenie obvodu FT232H s konfiguračnou pamäťou.....	62
A.3 Zapojenie TE0725 do RJ	63
A.4 Zapojenie rezervného zdroja video signálu	64
B doska plošných spojov	65
B.1 Vodivé vrstvy - TOP (vľavo) a BOT (vpravo)	65
B.2 Vnútna zemniaca vrstva č. 2 (vpravo) a vnútorná napájacia vrstva č. 15 (vľavo)	66
B.3 DPS osadenie súčiastok – horná vrstva (vľavo), spodná vrstva (vpravo)	67

ÚVOD DO PROBLEMATIKY

Prvý let modelu lietadla, zjazd na lyžiach zasneženou krajinou, či jazda na koni - to sú neopakovateľné momenty, pri ktorých nemáme vždy pri sebe kameramana, ochotného kvalitne tieto udalosti zaznamenať. Z tohto dôvodu bolo rozhodnuté vypracovať tento projekt, ktorý by umožňoval automatické natáčanie kamery na pohybujúci sa cieľ, ako je napríklad lyžiar na svahu, lietadlo alebo auto.

Práca je zameraná na návrh zameriavacieho systému, ktorý bude schopný sledovať užívateľom zvolený pohybujúci sa objekt, ktorý bude v centre zorného poľa akejkoľvek kamery. Ideálnym konceptom takéhoto zariadenia je “stlačiť a nestaraj sa”, čo by v praxi pre užívateľa znamenalo namieriť kameru na cieľ, stlačiť tlačidlo nahrávania a ďalej sa už o funkciu kamery nezaujímať.

Predmetom diplomovej práce je fyzická realizácia spojená s reálnym otestovaním zariadenia, ktoré bude v ďalej v texte nazývať v skrate SSPO (Systém pre Sledovanie Pohybujúcich sa Objektov) v teréne. Na obr. 1 je znázornený fyzický koncept SSPO.

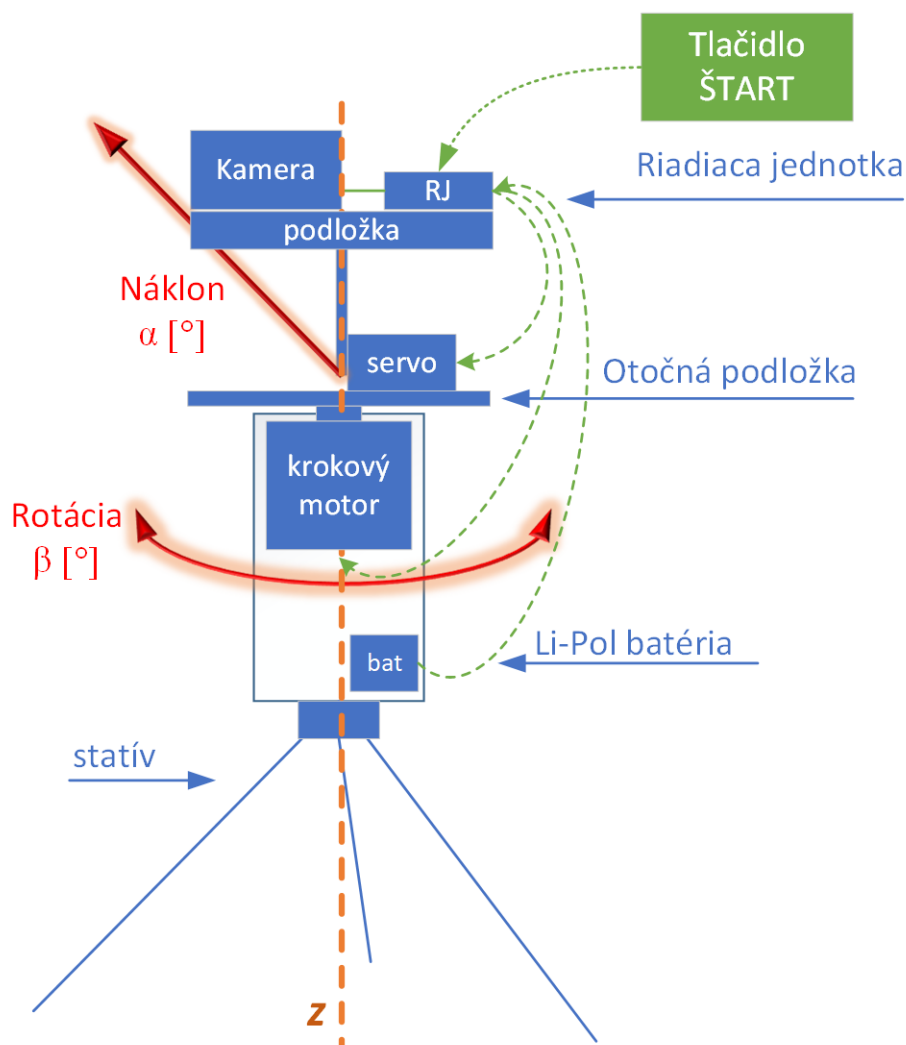


Obr. 1 Grafické znázornenie rozloženia SSPO v priestore

1 MECHANICKÝ KONCEPT SSPO

Táto stránka je venovaná predstaveniu základného konceptu mechanického usporiadania SSPO. Na obr. 2 je vidieť umiestnenie celého systému SSPO na kamerovom statíve. Konštrukcia SSPO je navrhnutá v CAD SolidWorks a následne vytlačená na 3D tlačiarňi z ABS (Akrylonitrilbutadiénstyrén) plastu. Konštrukciu SSPO je možné rozdeliť na niekoľko základných častí

- nepohyblivý obal krokového motora uchytený o vrchnú časť statívu,
- otočná podložka pripevnená o hriadeľ motora,
- servomotor,
- vrchná podložka s kamerou a riadiacou jednotkou,
- Li-Pol (Lítium-Polymér) batéria.



Obr. 2 Mechanické usporiadanie systému pre sledovanie pohybujúcich sa objektov

2 PREHLAD KOMERČNE DOSTUPNÝCH PRODUKTOV

Na súčasnom trhu sú pre zákazníkov k dispozícii výrobky, disponujúce podobnými parametrami ako systém, vytvorený v rámci predloženej diplomovej práce. Skonstruované zariadenie bude fungovať na inom princípe, ako komerčne dostupné produkty.

Nevýhodou produktov od spoločností SoloShot a Pixio je skutočnosť, že kamerovaný objekt musí neustále vysielat' svoju polohu určenú GPS (Global Positioning System) lokátorom a barometrom. Poloha je porovnávaná s polohou kamery a následne je vypočítavaný uhol natočenia. Táto metóda je preto silne obmedzená kvalitou rádiového spojenia medzi kamerou a vysielačím modulom umiestneným napríklad na ruke divadelného herca alebo lyžiara. Ďalšou nevýhodou je závislosť vysielačieho modulu na zdroji elektrickej energie a nutnosť kvalitného príjmu GPS signálu, ktorý v uzavretých priestoroch nie je k dispozícii. Z tohto dôvodu sa rozhodli spoločnosti implementovať dodatočný RFID (Radio Frequency Identification) systém s viacerými vysielačmi nazývanými aj ako beacons (ekvivalent satelitu v systéme GPS). Veľkou výhodou týchto systémov je ich nezávislosť na dobrej viditeľnosti objektu (hmla, sneženie a i.) a taktiež ich nezávislosť na veľkosti sledovaného objektu a na kvalite obrazového snímača alebo svetelnosti objektívu danej kamery. [1] [2]



Obr. 3 Soloshot 65 a PIXIO Zdroje: https://shop.moveensee.com/165-large_default/pixio-robotcameraman.jpg, [PIXIO-robot-cameraman.jpg](#)

Tab. 1 Prehľad základných parametrov komerčne dostupných výrobkov

	<i>Soloshot 65</i>	<i>PIXIO</i>
Cena bez DPH a dopravy [eur]	700	679
Výdrž batérie [hr]	-	2,5
Max. pozorovacia vzdialenosť [m]	609	700
Maximálna rýchlosť kamerovaného objektu na 10 m [km/h]	-	72

3 PRINCÍP SLEDOVANIA POHYBUJÚCICH SA OBJEKTOV Z VIDEOSEKVENCIE

Vytvorenie systému, ktorý by bol schopný sledovať pohybujúci sa objekt iba na základe obrazových informácií bol hlavným dôvodom vytvorenia tohto projektu. V tejto kapitole je čitateľovi predložený jeho koncept.

Hlavným predpokladom funkcie sledovania pohybujúceho sa objektu vo videosekvencii (séria snímok alebo fotiek vytvorených v časovej postupnosti) je možnosť získania zmien obrazovej informácie, ktorá vyjadruje pohyb. SSPO je možné potom rozdeliť z hľadiska implementácie základných algoritmov na nasledujúce bloky znázornené na obr. 4.

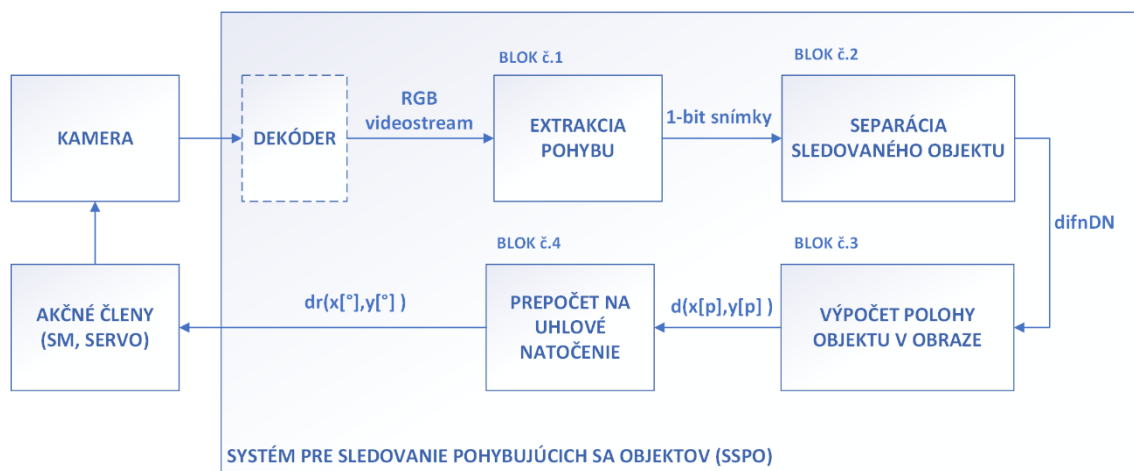
Blok č. 1 na obr. 4 obsahuje

- vytvorenie rozdielových zašumených snímok dif_{nN} popísaných v kapitole 6
- a ich filtráciu popísanú v kapitole 7.

Funkcia bloku č. 2: v prípade, že rozdielové snímky budú obsahovať niekoľko pohyblivých objektov, bude náš sledovaný predmet odseparovaný od ostatných na základe zaznamenávania poslednej polohy a špecifických znakov, ako sú napríklad rozmery.

Blok č. 3 bude ďalej obsahovať výpočet stredu C (*Center*) sledovaného objektu, ktorý bude súčasťou polohového vektora d udávajúceho pozíciu C vzhľadom k stredu obrazu F_c . Blok č. 3 je podrobne popísaný v kapitole č.8.

A funkciou bloku č.4 je transformácia vektora polohy d na vektor d_r do 3D (Three Dimensional) priestoru.



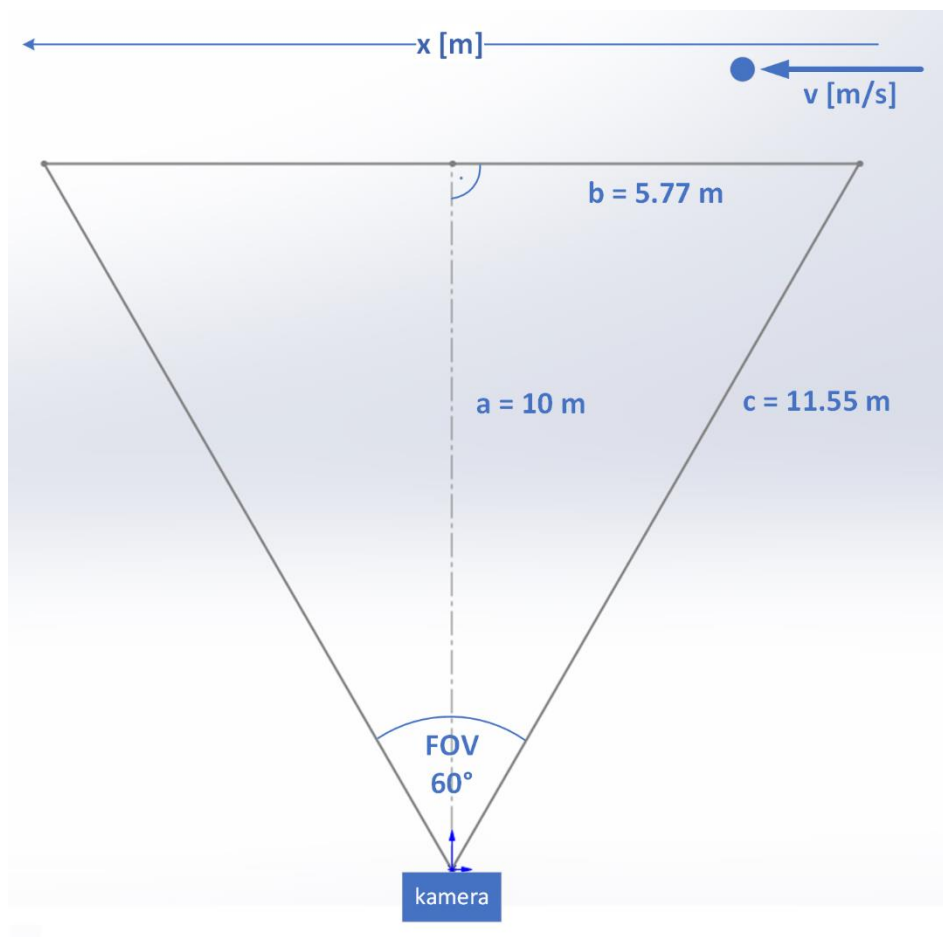
Obr. 4 Koncept systému SSPO

Dekódovanie výstupného formátu kamery tzn. fyzická implementácia vhodného rozhrania AV (Analog video), DVP (Digital Video Port) a iné je implementované vo vstupnom dekodéri popísanom v kapitole 9.5.1.

4 VÝPOČET MAXIMÁLNEJ RÝCHLOSTI OBJEKTU PRE NAŠE RIEŠENIE

Rýchlosť pohybu snímaného objektu je jedným z parametrov, ktoré silne limitujú aplikačné možnosti nasadenia systému v teréne. Táto časť sa preto podrobne venuje jej výpočtu.

Rýchlosť je fyzikálna veličina vyjadrujúca zmenu polohy d v pevne ohraničenom časovom intervale t . V našom prípade bude d označovať zmenu polohy sledovaného objektu na obr. 5 v ose x . Na nasledujúcom obrázku obr. 5 je ilustrovaná základná myšlienka pre získanie maximálnej rýchlosti v_{max} , ktorá vychádza z geometrických pomerov v pravouhlom trojuholníku a, b, c .



Obr. 5 Grafické znázornenie relácie zorného poľa kamery s pohybujúcim sa cieľom

Predpokladaná približnú vzdialenosť kamery c od sledovaného objektu. Rýchlosť tohto objektu vzhľadom ku kamere bude najväčšia v prípade, že trajektória pohybu bude kolmá k priamke a , ktorá definuje smer natočenia kamery v 2D (Two Dimensional) priestore. Pre výpočet v_{max} je nevyhnutné zistiť maximálnu kamerou zachytiteľnú zmenu polohy l

za jednu snímkovaciu periódu. Maximálna zmena polohy, ktorú bude kamera schopná zachytiť je závislá iba na tangense uhla FOV (Field of View) a priamke a . Na nasledujúcej strane sú uvedené matematické výpočty k obr. 5.

Výpočet maximálne možnej zachytiteľnej zmeny polohy je

$$l = 2 \times b \text{ [m]}, \quad (1)$$

$$b = \tan\left(\frac{FOV}{2}\right) \times a \text{ [m]}, \quad (2)$$

kde b je polovičná trajektória objektu premietnutá do zorného poľa, l je maximálna zachytiteľná zmena polohy objektu na vzdialenosť 10 metrov a FOV je uhol zorného poľa kamery.

Výsledná rýchlosť je potom vyjadrená ako

$$v_{\max} = \frac{l \times f_{cam}}{3 \times div} = \frac{11,54 \times 15}{3} = 57,7 \text{ [ms}^{-1}\text{]} \quad (3)$$

kde v_{\max} je maximálna rýchlosť kamerovaného objektu, l je maximálna zachytiteľná zmena polohy objektu na vzdialenosť 10 metrov, f_{cam} je snímkovacia frekvencia kamery a div deliaci pomer určujúci reálnu snímkovaciu frekvenciu použitú pre sledovací algoritmus.

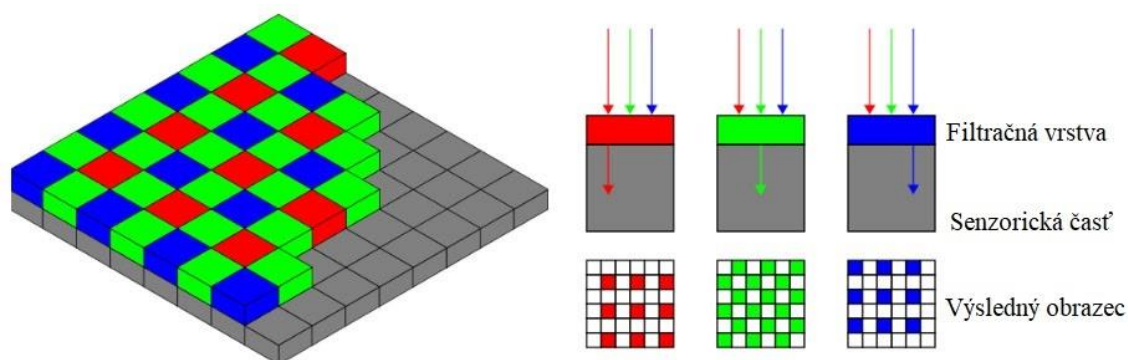
5 DEFINÍCIA OBRAZOVÝCH FORMÁTOV

Táto kapitola je venovaná základným pojmom v oblasti bitmapovej techniky zobrazovania vizuálnych informácií.

Digitálny obraz je možné definovať ako množina diskretných bodov v dvojrozmernom priestore, ktorá má pevne definovanú dĺžku a šírku, označovanú ako rozlíšenie v horizontálnom a vertikálnom smere. Každý obrazový bod je označovaný ako pixel, udávajúci informáciu o farbe a jej intenzite v danom mieste. V počítačovej technike je vo väčšine prípadov používaný farebný formát 32-bit ARGB, kde *A* je Alpha alebo aj priesvitnosť, *R* je červená, *G* je zelená a *B* je modrá komponenta pixelu v 8 bitovom formáte. Každá z komponent môže nadobúdať 256 hodnôt. Zdrojom obrazovej informácie je kamera, preto výstup neobsahuje Alpha zložku. Veľmi výhodné je použitie bezkompresného formátu RAW 24-bit RGB, ktorý je vhodný pre ďalšie spracovanie a vyhodnocovanie. Jeho jedinou nevýhodou je, že v prípade ukladania zaberá maximálny možný počet bitov. [3]

Bežná štruktúra obrazového snímača v technológii CMOS je zložená z jednej bunky. Každá bunka obsahuje fotodiódu zapojenú v závernom režime a nábojový zosilňovač. Nad touto štruktúrou je umiestnená filtračná vrstva svetla v Bayerovom usporiadaní zobrazená na obr. 6. Jeden výsledný pixel je tvorený štvoricou subpixelov pre každú farbu. Jeden subpixel je venovaný červenej farbe a jeden modrej farbe. Zvyšná dvojica je určená pre zelenú farbu z dôvodu väčšej citlivosti ľudského oka na spektrum v okolí 500 nm. Poslednou vrstvou sú optické šošovky slúžiace na zaostrenie svetla do jedného jasne ohraničeného priestoru senzorickej časti. [4]

Výstup CMOS snímača je reprezentovaný hodnotami intenzít farieb RGB idúcich v sérii po jednotlivých riadkoch v usporiadaní znázornenom na obr. 6.



Obr. 6 Fyzická štruktúra obrazového snímača a jeho výstupný dátový formát [4]

Ďalej bude v tomto projekte využívaná iba jasová (luminescenčná) zložka farebného modelu YCbCr, kde Y udáva hodnotu jasu, Cb , Cr sú modrá a červená rozdielová chrominančná zložka. Výpočet jasovej zložky z RGB modelu Y je definovaný ako

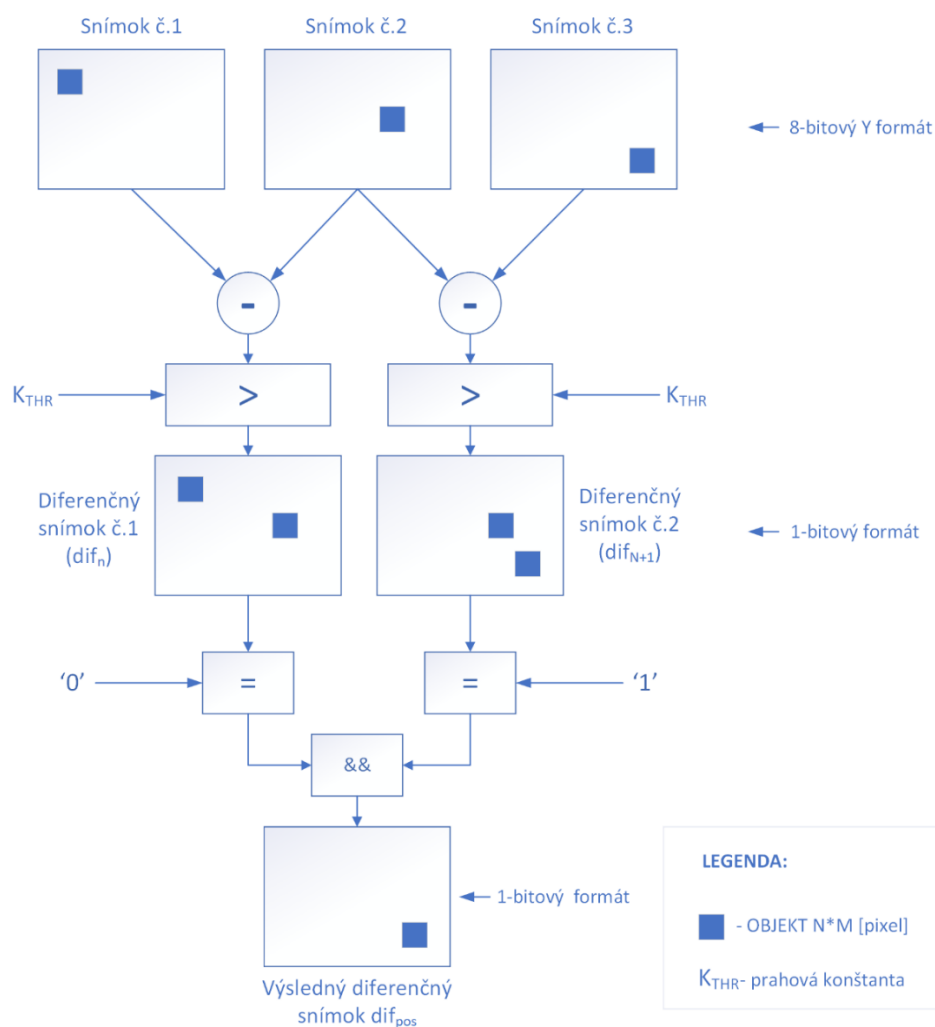
$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B [-], \quad (4)$$

kde R , G a B sú farebné komponenty. [5]

6 PRINCÍP EXTRAKCIE POHYBÚCICH SA OBJEKTOV Z VIDEO SEKVENCIE

Táto kapitola sa venuje princípu vytvorenia diferenčného alebo rozdielového snímku, ktorý poskytuje informáciu o pohybe v snímanom zornom poli zo série snímok získaných z kamerového snímača. Tieto snímky budú ďalej vyhodnocované na získanie polohy pohybujúceho sa objektu.

Na úvod treba zdôrazniť, že hlavnou podmienkou správnej funkcie nasledujúcej metódy je nehybnosť obrazového snímača počas získavania snímok z kamery. Ak by nastal pohyb kamery počas ich získavania, algoritmus by nebol schopný oddeliť statické pozadie od pohybujúceho objektu a algoritmus by nefungoval korektne. Tento problém je vyriešený vypínaním procesu znázorneného na nasledujúcom obrázku počas pohybov kamery.



Obr. 7 Znázornenie procesu extrakcie diferenčného snímku dif_{pos}

Na predchádzajúcej strane na obr. 7 je znázornený pyramídový algoritmus extrakcie pohybu v sérii nekomprimovaných snímok. V prvej fáze je potrebné vytvoriť diferenčný snímok dif_n odčítaním snímku č.1 a snímku č.2 porovnaním výsledku odčítania s experimentálne určenou hraničnou konštantou K_{THR} . Ak bude výsledok odčítania väčší ako K_{THR} budú nastavené jednotlivé bity v bitovom poli dif_n . Rovnaký postup je aplikovaný na snímok č.2 a č.3. Po získaní dif_n a dif_{n+1} je potrebné vyextrahovať poslednú zaznamenanú pozíciu objektu premietnutú do výsledného diferenčného snímku dif_{pos} . V ďalšom procese bude tento diferenčný snímok spracovávaný ako dvojrozmerné bitové pole z ktorého budú po odfiltrovaní šumu a nežiaducich zmien vypočítané súradnice sledovaného objektu. [6] [7]

Vyššie uvedený princíp je vhodný pre implementáciu do zariadenia, ktoré by umožnilo paralelizmus na úrovni získavania a spracovávania. Na tieto typy úloh sa ako vhodné javí FPGA (Field Programmable Gate Array). V FPGA sa dajú jednoduchým spôsobom implementovať samostatné bloky (akcelerátory) pre získavanie a spracovávanie obrazových informácií v reálnom čase.

7 ODSTRÁNENIE ŠUMU Z ROZDIELOVÉHO SNÍMKU

Nasledujúce riadky sú zamerané na odstránenie šumu a malých obrazových zmien v rozdielovom snímku popísaného v predchádzajúcej kapitole.

7.1 Zdroje šumu na vstupných snímkach

Základným pôvodcom šumu na obrázku je tepelný šum, spôsobený pohybom elektrónov v kryštálovej mriežke polovodičov. S rastúcou teplotou stúpa frekvencia kmitania kryštalografickej štruktúry polovodiča, brániaca pohybu elektrónov. Následkom je zvýšenie šumového pozadia s konštantnou spektrálnou hustotou.

Ďalším zdrojom degradácie obrazu je tzv. temný šum, ktorý vzniká tepelnou generáciou voľného náboja v depletičnej vrstve *PN* priechodu.

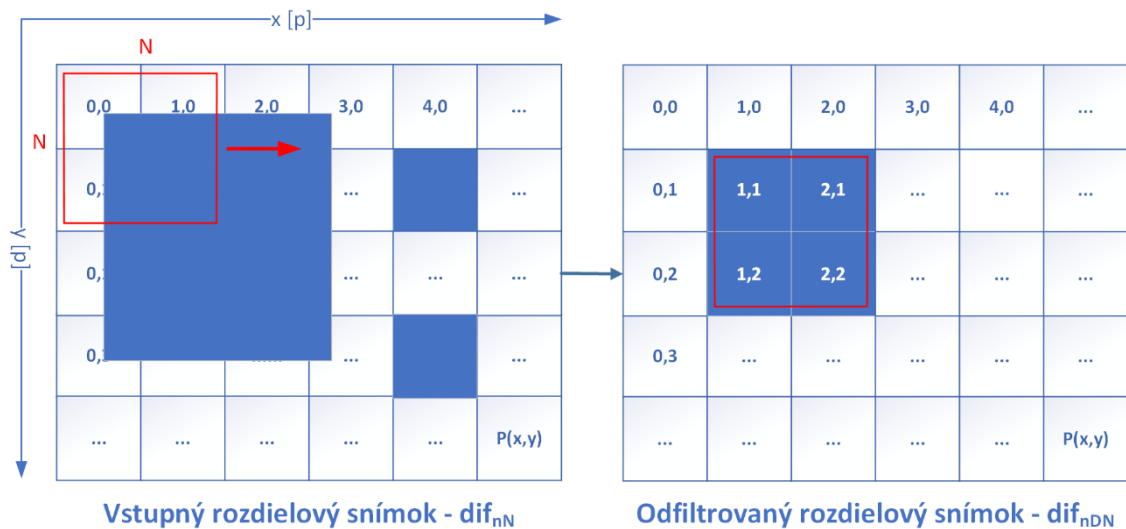
Je možné si všimnúť, že fotky vytvorené pri nízkej hodnote osvetlenia majú vyššiu zašumenosť. Je to spôsobené tým, že pri nižšej svetelnosti je potrebné signál z obrazového snímača viac zosilniť, ale tým sa logicky zosilní aj úroveň šumového pozadia. [8] [9]

Na záver je treba podotknúť, že všetky komponenty v elektrickom obvode šumia a majú vplyv na výsledný obraz. Najvýznamnejší dopad na výslednú kvalitu snímku majú analógovo-digitálne prevodníky, ktoré svojím skreslením, nízkym odstupom šumu od užitočného signálu alebo prípadnou nelineárnou prevodnou charakteristikou môžu výsledný obraz znehodnotiť.

7.2 Vplyv vstupného šumu na rozdielové snímky

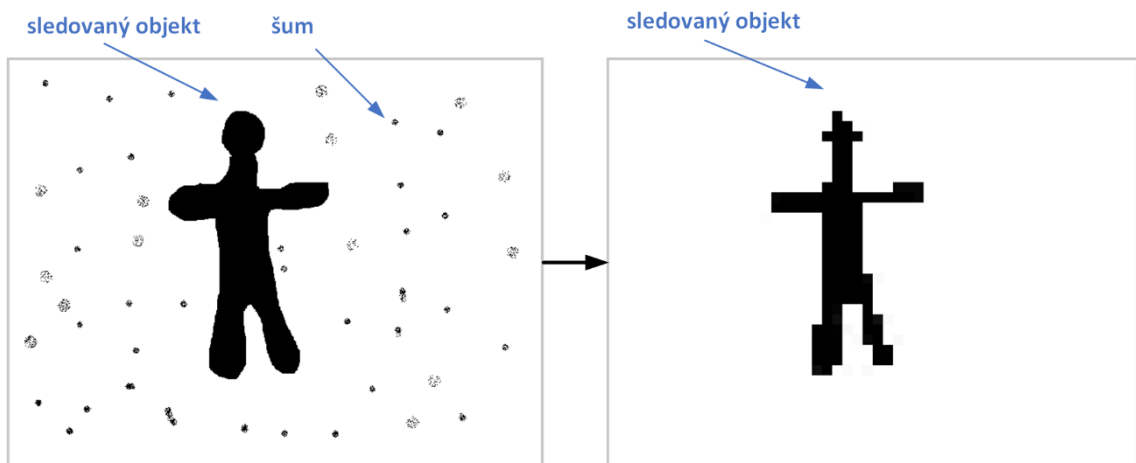
Odstránenie šumu a malých zmien pohybu ako sú napríklad lístie alebo tráva vo vetre v rozdielových snímkach dif_{pos} popísaných v kapitole 6 je nevyhnutné pre ich ďalšie spracovávanie.

Odstránenie objektov menšej veľkosti ako je filtračné okno s rozmermi N, N pixelov je veľmi jednoduché. Princípom je prechádzať filtračným okienkom bod po bode celým obrazom dif_{nN} a v prípade, že sú pod ním všetky body (na obr. č. 8 sú to 4 body) nastavené do log.1, bude celé okienko prekopírované do výsledného odfiltrovaného snímku dif_{nDN} . Na nasledujúcej strane je tento proces pre lepšie pochopenie graficky znázornený.



Obr. 8 Princíp okienkovej filtrácie

Na obr.9 je znázornený očakávaný výsledok filtračného mechanizmu z obr.8.



Obr. 9 Grafické znázornenia okienkovej filtrácie vstupného zašumeného snímku dif_{nN} na výsledný dif_{ndN} snímok vo formáte jednobitového dvojrozmerného poľa

Nevýhodou vyššie uvedeného algoritmu je jeho veľká výpočtová náročnosť. V prípade prístupu do jediného pamäťového miesta sa dá počet čítaní a zápisu pre odfiltrovanie celého obrazu vyjadriť ako

$$N_{max} = (x - 1) \times (y - 1) \times 2 \quad (5)$$

kde x a y sú rozlíšenia vstupného zašumeného snímku dif_{nN} v horizontálnom a vertikálnom smere.

Pri rozlíšení SVGA (Super Video Graphic Array) 800x600 pixelov bude počet prístupov do pamäte

$$N_{max} = (800 - 1) \times (600 - 1) \times 2 = 957\,202 \quad (6)$$

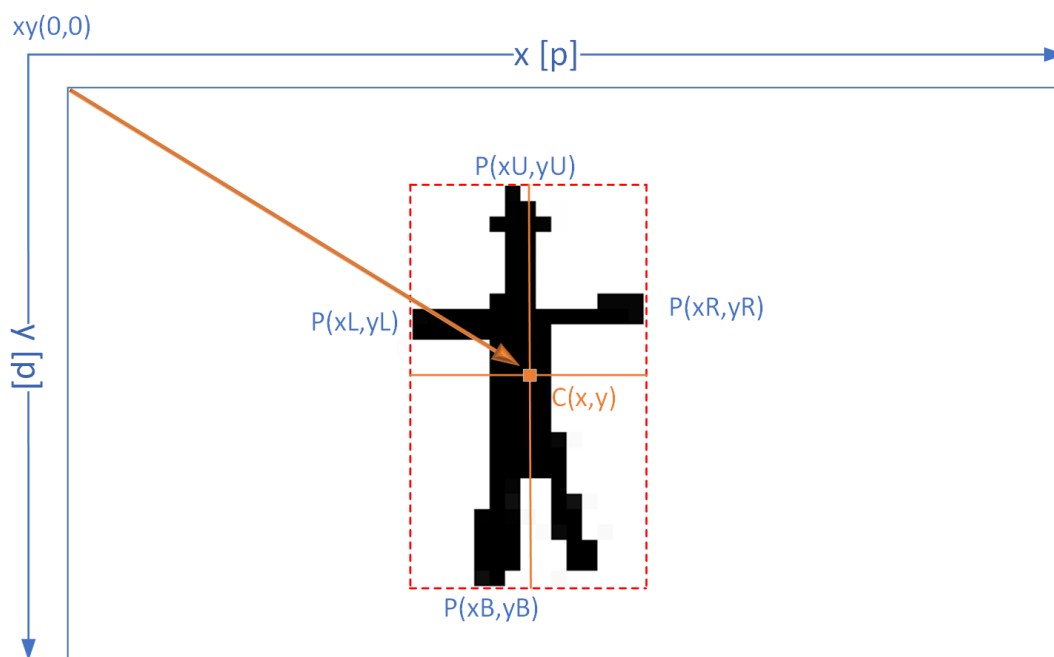
čo pri štandardnej RAM(Random Access Memory) DDR2 (Double Data Rate) jedno-portovej pamäti s 8bitovým komunikačným rozhraním taktovaným na frekvenciu f_{RAM} zaberie približne čas

$$T_{max} \cong \frac{N_{max}}{2 \times f_{RAM}} = \frac{1\,117\,002}{2 \times 133\,000\,000} = 3,5\,ms \quad (7)$$

, ktorý musí byť menší ako snímkovacia frekvencia kamery. V ďalších kapitolách praktickej realizácie bude podrobnejšie uvedené riešenie tohto problému rozdelením obrazu do viacerých pamäťových miest alebo spracovávaním obrazu za súčasného získavania snímok z kamery.

8 VÝPOČET STREDU OBJEKTU Z MNOŽINY BODOV

Získanie jedinečného diskrétného bodu $C_{(x,y)}$, ktorý udáva stred sledovaného objektu je v 2D súradnicovom systéme s počiatkom v pravom hornom rohu je nevyhnutné pre ďalšiu navigáciu kamery. Na obr. 10 je graficky znázornený príklad výpočtu stredy objektu v odfiltrovanom diferenčnom snímku $diff_{pos}$ popísaného v predchádzajúcej kapitole. Princípom je získanie hraničných bodov P v súradnicovom systéme x, y .



Obr. 10 Znázornenie výpočtu približného stredy sledovaného objektu

Postup výpočtu $C_{(x,y)}$ premietnutého do 2D priestoru je riešený nasledujúcim postupom:

1. Získanie súradníc najvyššie a najnižšie položeného bodu $P(xU,yU)$, $P(xB,yB)$.
2. Získanie súradníc ľavého $P(xL,yL)$ a pravého okraja $P(xR,yR)$ objektu.

Následne je možné jednoduchým výpočtom určiť stred

$$C(x, y) = \left[\frac{P(xR) - P(xL)}{2}, \frac{P(yB) - P(yU)}{2} \right] [p, p], \quad (8)$$

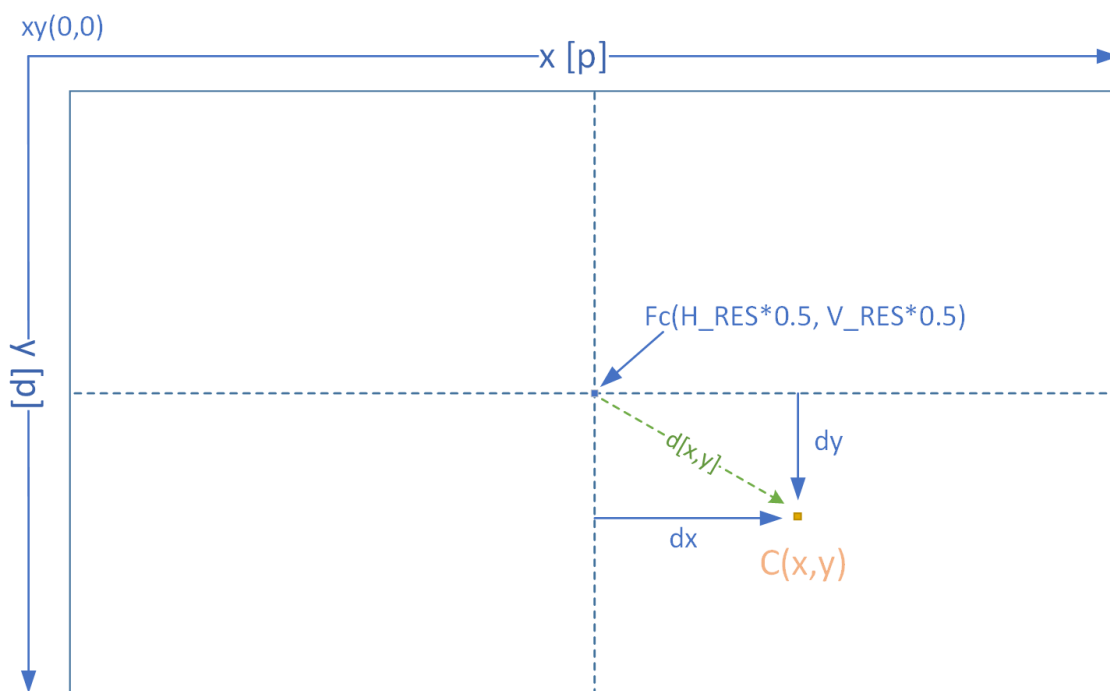
kde $P(xU, yU)$ upper point (Najvyššie položený bod) a $P(xB, yB)$ najnižšie položený bod je množina bodov definujúce výšku a šírku sledovaného objektu v jednotkách pixelov.

V ďalšom kroku je potrebné vyjadriť vzťah medzi stredom obrazu F_c (smer mierenia kamery) s aktuálnou pozíciou objektu $C(x,y)$. Pre výpočet vektoru aktuálnej vzdialenosti $d(x,y)$ platí

$$d(x,y) = C(x,y) - F_c(x,y) [p,p], \quad (9)$$

$$F_c(x,y) = \frac{H_{RES}}{2}, \frac{V_{RES}}{2} [p,p], \quad (10)$$

kde H_{res} a V_{res} sú horizontálne a vertikálne rozlíšenie kamerového snímača.



Obr. 11 Znáznornenie vzdialenosti sledovaného objektu od stredu obrazu v pixeloch

9 PRAKTICKÁ REALIZÁCIA SSPO

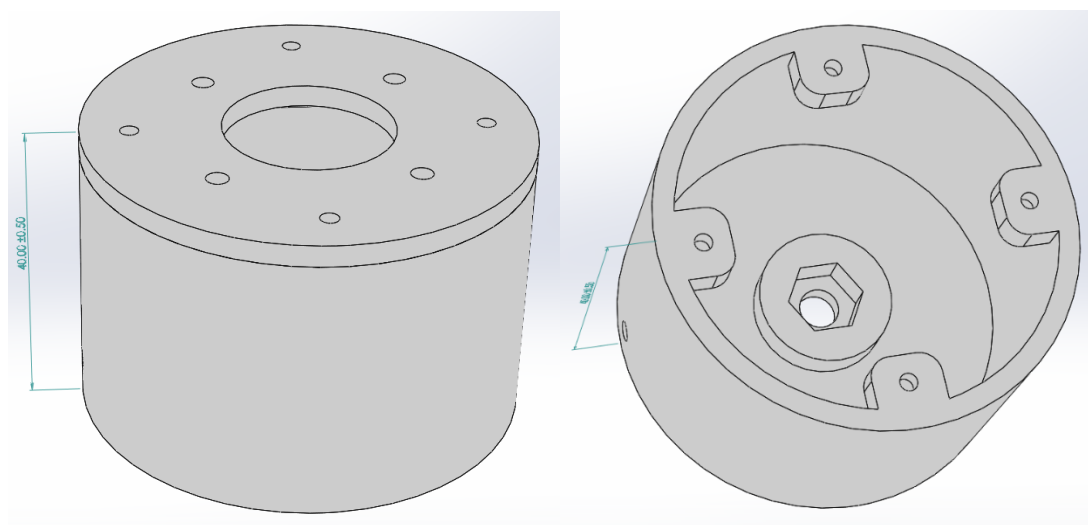
Táto časť práce sa zaoberá návrhom SSPO na fyzickej úrovni, ktorý bol v rámci diplomovej práce aj prakticky realizovaný a otestovaný v teréne. Kapitola je rozdelená do niekoľkých častí, ktoré by mali čitateľa postupne naviesť na správne riešenie zadaného problému.

9.1 Návrh mechanicko-motorickej časti

V kapitole 1 bol popísaný mechanický koncept SSPO. Táto časť je venovaná výberu adekvátneho krokového motora a mechanickej konštrukcii SSPO.

Otáčanie okolo osi z z obr. 2 v kapitole 1 je realizované bipolárnym krokovým motorom POLOLU 1207. Tento motorček má hodnotu jedného kroku len $1,8^\circ$, ktorú je možné ešte znížiť na $0,9^\circ$ simultánnym budením oboch vinutí tzv. half-stepping. Jeho celkový nominálny prúdový odber pri napájacom napätí 7,4 V je približne 0,5 A. [10]

Na obr. 12 sú znázornené dva pohľady na 3D model statického obalu krokového motora POLOLU 1207.



Obr. 12 Pohľad na 3D model obalu krokového motora vo vývojovom prostredí CAD SolidWorks2018

Ďalšie pohyblivé časti ako sú otočná podložka a naklápacia podložka z obrázku č. 2 so servomotorom sú navrhnuté a vytlačené na 3D tlačiarňi ENDER-3.

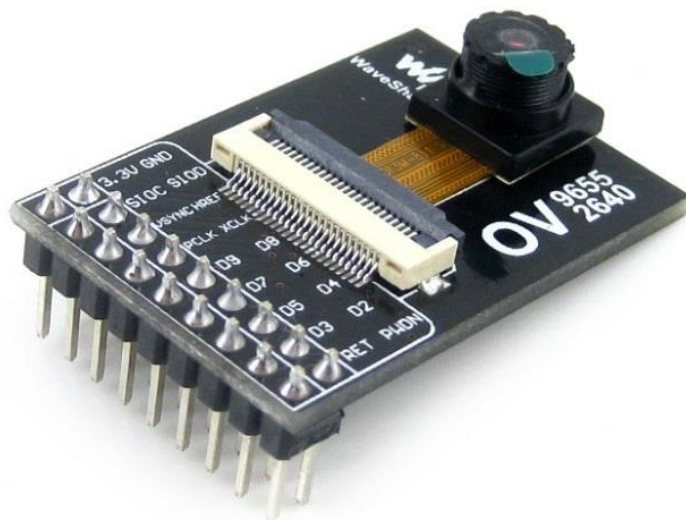
9.2 Výber testovacej kamery

Na trhu sa nachádza niekoľko kamerových modulov, ktorých výstupy sa dajú v FPGA jednoducho spracovávať.

Cenovo dostupný kamerový čip alebo aj SoC (System on Chip) od spoločnosti OmniVision s typovým označením OV2640 CAMERACHIP™ je ideálnym kandidátom na implementáciu do systému. Snímač disponuje rozlíšením UXGA s obnovovacou frekvenciou 15 Hz, ktorá je plne postačujúca. Ďalej je na čipe implementovaný DSP (Digital Signal Processor) pre spracovávanie obrazu do štandardného výstupného formátu s možnosťou kompresie, pod vzorkovania na nižšie rozlíšenie, automatické nastavenie expozície AEG (Automatic Exposure Gain) alebo zrkadlenie obrazu. [11]

Na čipe je ďalej využité sériové komunikačné rozhranie SCCB (Serial Camera Control Bus), ktoré je ekvivalentom k štandardu I²C (Inter-Integrated Circuit). Prenos videosignálu prebieha cez DVP (Digital Video Port). Výstupný formát obrazových dát z modulu je nastaviteľný poslaním konfiguračnej sekvencie cez rozhranie SCCB. [11]

Pre zákazníkov je na trhu k dispozícii nadstavbový modul obsahujúci tento čip pripevnený na DPS od výrobcu WAVESHARE s označením 8532. Modul obsahuje samotný kamerový čip, lineárne stabilizátory napätia pre jeho napájanie analógovej a digitálnej časti a filtračné obvody. Pre jednoduchú komunikáciu s okolím je na doske pripájaná kolíková lišta. Kvalita spracovania na štandardnej úrovni. [12]



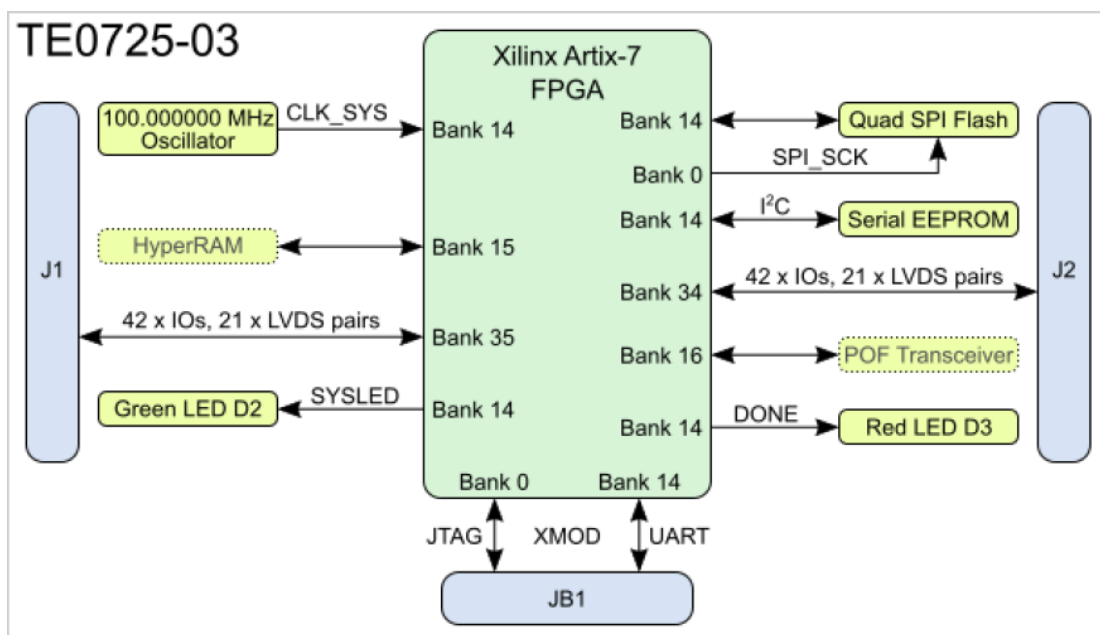
Obr. 13 Kamerový modul WAVESHARE 8532 s čipom OV 2640 Zdroj: https://www.waveshare.com/.../ov2640-camera-board_1_1_5.jpg

9.3 Výber vhodnej realizačnej platformy a jej parametre

Ako už bolo spomenuté ideálnou platformou pre paralelné spracovávanie obrazu je programovateľné hradlové pole FPGA. Z dôvodu dobrých skúseností s FPGA od spoločnosti Xilinx sa bude táto časť textu zameriavať iba na ich cenovo dostupné varianty.

Spoločnosť Xilinx je popredný výrobca FPGA pre komerčnú sféru trhu. V posledných rokoch boli predstavené nové architektúry siedmej generácie Artix, Virtex, Kintex a Spartan postavené na technológii s dĺžkou kanálu tranzistora MOS iba 28 nm. Cenovo a využiteľnou plochou čipu je pre projekt dostupná architektúra Artix, ostatné sú vo vyššej cenovej relácii a architektúra Spartan siedmej generácie je nedostačujúca. Všetky FPGA vo vyššie uvedených architektúrach sú zapuzdrené do puzdier typu BGA (Ball Grid Array), ktoré sú v amatérskych podmienkach ťažko pájkovateľné. [13]

Z tohto dôvodu bolo zvolené vhodné modulárne riešenie, ktoré značne urýchli vývoj celého zariadenia. Cenovo prijateľný je modul TE0725 verzia 3 od výrobcu Trenz Electronic GmbH. Modul obsahuje DRAM (Dynamic Random Access Memory), ktorá bude využitá pre ukladanie veľkého objemu medzivýsledkov. Na nasledujúcom obrázku je znázornený blokový diagram modulu.



Obr. 14 Blokový diagram modulu TE0275-3 Zdroj: <https://wiki.trenz-electronic.de/download/...>

DPS modul TE0275-3 obsahuje všetky nevyhnutné komponenty pre realizáciu SSPO vrátane pamäte EEPROM pre uloženie bitstreamu a konfigurácie FPGA, JTAG (Joint Test Action Group) rozhranie pre programovanie a debug FPGA a kolíkové lišty J1, J2 pre jednoduché pripojenie do základnej DPS riadiacej jednotky. [14]

9.4 Koncept základnej dosky plošných spojov pre RJ

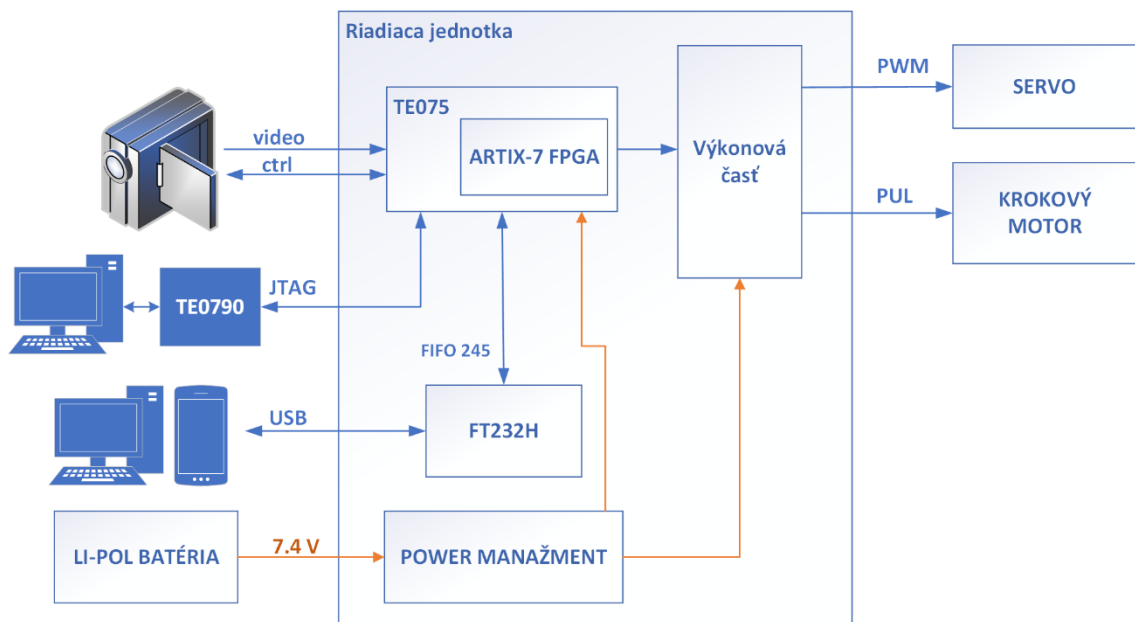
Táto časť textu popisuje návrh riadiacej jednotky celého systému SSPO na blokovej úrovni, ktorý bude pretvorený do DPS. Správne navrhnutý koncept RJ bude základom pre funkčné zariadenie.

Obsahom riadiacej jednotky z obr. 12 je

- kamerový modul OV 2640 popísaný v kapitole 9.2,
- modul Trenz Microeletronic TE0725 s FPGA 7 Artix-7 XC7A35T-2CSG324,
- čip pre prevod USB 2.0 (Universal Serial Bus) rozhrania na paralelne vysokorychlostné synchrónne FT FIFO 245,
- výstupný výkonový stupeň pre riadenie polohy serva a krokového motora,
- napájacie a filtračné obvody s Li-Pol batériou.

Ďalej je pre validáciu SSPO použitý

- programátor Trenz JTAG TE0790,
- počítač s aplikáciou postavenej na platforme .NET C# pre validáciu celého zariadenia,
- počítač s vývojovým kitom Vivado Xilinx.

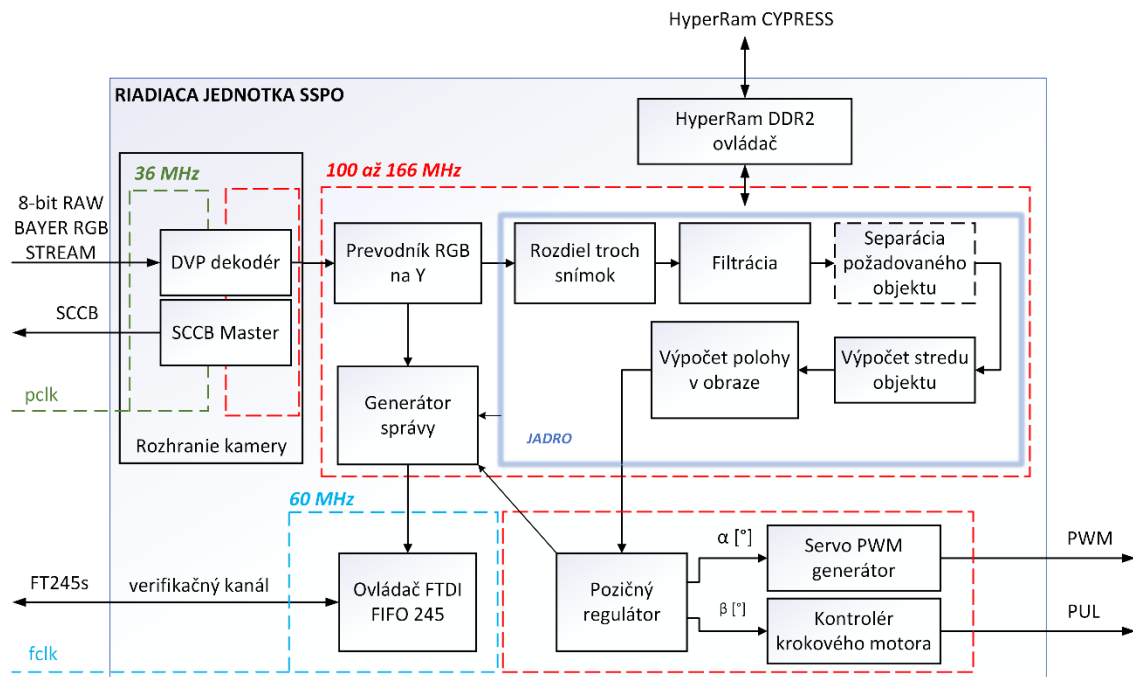


Obr. 15 Blokové znázornenie RJ v systéme SSPO

9.5 Koncept RJ v programovateľnom hradlovom poli FPGA

Tento popis praktickej realizácie v FPGA sa zaoberá jej konceptom na blokovej a RTL (Register Transfer Logic) úrovni. Návrh je vytvorený v najnovšom vývojovom štúdiu Xilinx Vivado 2018 v popisnom jazyku VHDL (VHSIC Hardware Description Language). Bližšie špecifikácie jednotlivých blokov sú uvedené d'alších kapitolách.

Na obr. 16 je znázornený dočasný koncept RJ s primárnymi vstupmi a výstupmi, ktoré sú fyzicky vyvedené z FPGA. V rámci diplomovej práce bola RJ rozšírená o niekoľko ďalších entít, ktoré sú nevyhnutné pre správnu funkciu SSPO.



Obr. 16 Koncept RJ v hradlovom poli FPGA Artix-7

Riadiacu jednotku je zatiaľ možné rozdeliť z hľadiska funkčnosti na nasledujúce bloky:

- Rozhranie kamery s DVP dekodérom s SCCB rozhraním,
- prevodník RGB na Y zložku obrazu,
- jadro pre získanie pozície objektu v obraze,
- HyperRam DDR2 ovládač,
- pozičný regulátor pre prepočet pozície objektu v obraze na uhlové natočenia,
- servo PWM generátor alebo kontrolér,
- kontrolér krokového motora.

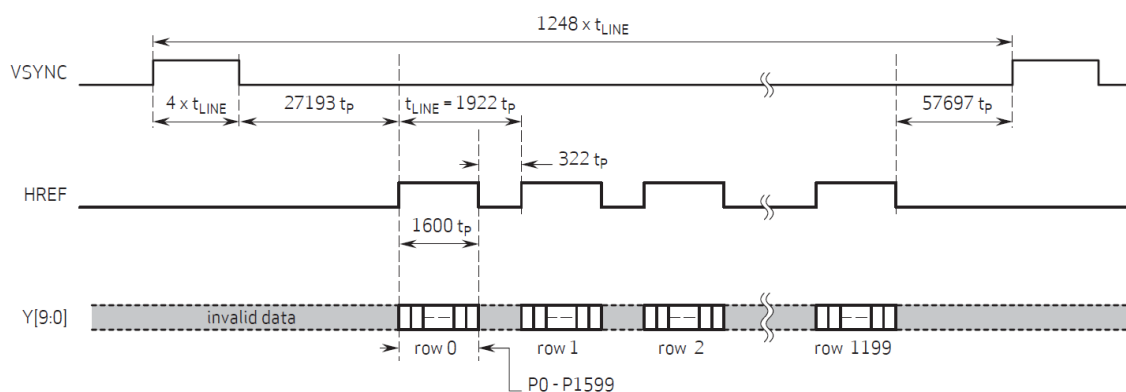
V nasledujúcich kapitolách je podrobnejšie rozobraná ich funkcia.

9.5.1 DVP dekodér

Úlohou DVP dekodéra je dekodovanie výstupných dát z kamery OV2640 idúcich v sérii podľa Bayerovho obrazca na paralelný 24-bitový formát RGB. Pre každú farbu R, G a B bude potom na výstupe v jednom okamihu k dispozícii 8 bitov. Vstupom do DVP dekodéra sú ďalej synchronizačné signály pre vertikálnu a horizontálnu synchronizáciu VSYNC, HREF. [11]

Na obrázku obr. 17 je znázornený typický priebeh dát a synchronizačných signálov z DVP s nastavením rozlíšenia na formát UXGA. Výstupné dáta z kamery Y[9:0] sú zakódované v Bayerovom kóde (popísanom v kapitole 5), ktorých postupnosť v čase je znázornená na obr. 18. Ako je možné vidieť jednotlivé zložky pixelov prichádzajú s počiatkom v ľavom hornom rohu v poradí pre prvý nepárny riadok: B₀, G₀, B₁, G₀ až G₁₅₉₉. Pre párne riadky vo formáte: G₀, R₀, G₁, R₀ až G₁₅₉₉. Rovnaký formát dát Y sa strieda aj pre ďalšie riadky. Z toho vyplýva, že na vytvorenie jedného kompletného pixelu RGB je potrebné uchovávať hodnoty predchádzajúceho riadku napríklad do blokových pamätí RAM. Ďalej je nevyhnutné spomenúť, že dáta Y sú validné iba ak je HREF v log.1. V SSPO bude využitý osembitový rozsah tzn., že sa zahodia spodné dva bity.

Obrazové dáta Y na vstupe DVP sú validné s príchodom nábežnej hrany hodinového signálu pclk, ktorý je v designe využitý pre ich zápis do synchronizačnej pamäte FIFO (First in First Out) Sync_FIFO. Prijatie jedného kompletného snímku vyžaduje synchronizáciu stavového automatu Control_FSM na začiatok prvého riadku detekciou nábežnej hrany signálu VSYNC. Ďalej je potrebné uchovávať prijatý počet bodov v jednom riadku čítačom color_addr, ktorý je nulovaný s príchodom posledného pixelu v riadku.



Obr. 17 Priebeh synchronizačných signálov VSYNC, HREF vzhľadom k dátam Y Zdroj:

https://www.uctronics.com/download/cam_module/OV2640DS.pdf

R/C	0	1	2	3	...	1598	1599
0	B _{0,0}	G _{0,1}	B _{0,2}	G _{0,3}	...	B _{0,1598}	G _{0,1599}
1	G _{1,0}	R _{1,1}	G _{1,2}	R _{1,3}	...	G _{1,1598}	R _{1,1599}
2	B _{2,0}	G _{2,1}	B _{2,2}	G _{2,3}	...	B _{2,1598}	G _{2,1599}
3	G _{3,0}	R _{3,1}	G _{3,2}	R _{3,3}	...	G _{3,1598}	R _{3,1599}
⋮					⋮		
1198	B _{1198,0}	G _{1198,1}	B _{1198,2}	G _{1198,3}	...	B _{1198,1598}	G _{1198,1599}
1199	G _{1199,0}	R _{1199,1}	G _{1199,2}	R _{1199,3}	...	G _{1199,1598}	R _{1199,1599}

Obr. 18 Výstupný formát dát UXGA RGB RAW z kamery OV2640 Zdroj:
https://www.uctronics.com/download/cam_module/OV2640DS.pdf

Principiálne je možné RTL návrh DVP dekodéru zverejnený na nasledujúcej strane rozdeliť na:

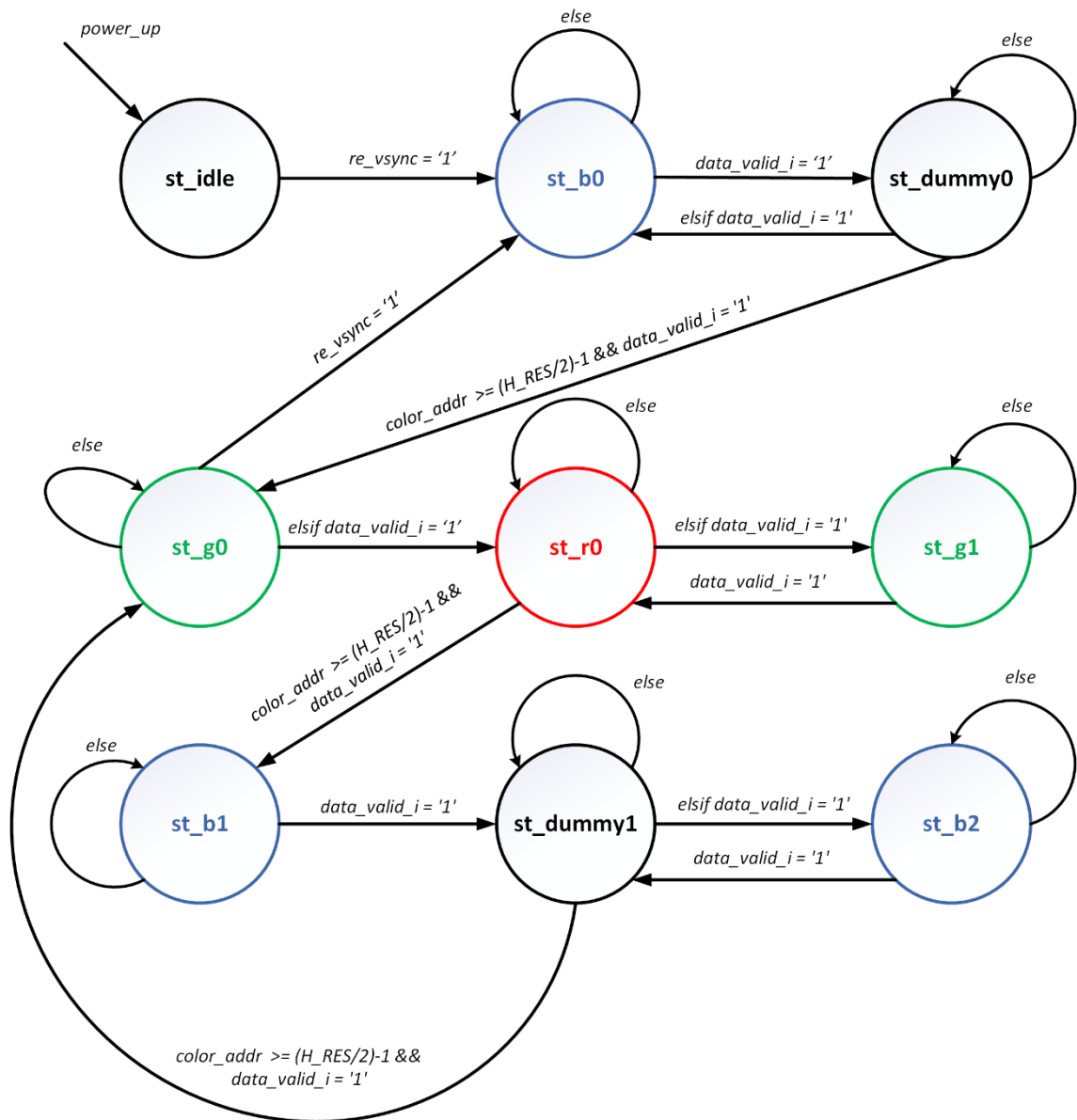
- Čítanie dát z asynchrónnej hodinovej domény pclk na obr. 19 (zelená sekcia) pomocou synchronizačnej pamäte FIFO implementovanej pomocou IP FIFO generátoru.
- Konečný stavový automat Control_FSM.vhd typu Mealy, ktorého úlohou je riadenie zápisu, čítania blokových pamätí RAM v FPGA a multiplexovania ich výstupu s výstupom synchronizačnej FIFO (First in First Out) Sync_FIFO.

Na nasledujúcej tabuľke je vyjadrená funkcia výstupného dekodéru Control_FSM.vhd v závislosti na aktuálnom stave present_state a kontrolnom signále data_valid_i.

Tab. 2 Hodnoty závislých výstupov Control_FSM.vhd na aktuálnom stave a vstupe data_valid_i

Hodnota vstupov/Aktuálny stav	st_idle
data_valid = 0	
data_valid = 1	
Nezávislé výstupy	color_addr_ce <= '0'; ram_wr_en <= "000"; color_addr_red <= color_addr; out_en <= '0';
Hodnota vstupov/Aktuálny stav	st_b0
data_valid = 0	ram_wr_en <= "000";
data_valid = 1	ram_wr_en <= c_BLUE;
Nezávislé výstupy	color_addr_ce <= '0'; color_addr_red <= color_addr; out_en <= '0';
Hodnota vstupov/Aktuálny stav	st_dummy0
data_valid = 0	color_addr_ce <= '0';
data_valid = 1	color_addr_ce <= '1'; ram_wr_en <= c_GREEN
Nezávislé výstupy	ram_wr_en <= "000"; color_addr_red <= color_addr; out_en <= '0';
Hodnota vstupov/Aktuálny stav	st_g0
data_valid = 0	ram_wr_en <= "000";
data_valid = 1	ram_wr_en <= c_GREEN
Nezávislé výstupy	color_addr_ce <= '0'; color_addr_red <= color_addr; out_en <= '0';
Hodnota vstupov/Aktuálny stav	st_r0
data_valid = 0	ram_wr_en <= "000"; color_addr_ce <= '0'; out_en <= '0';
data_valid = 1	ram_wr_en <= c_RED; color_addr_ce <= '1'; out_en <= '1';
Nezávislé výstupy	color_addr_red <= color_addr;
Hodnota vstupov/Aktuálny stav	st_g1

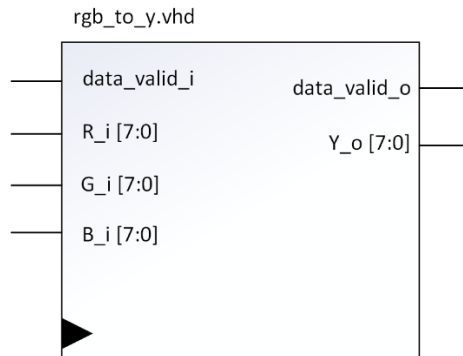
data_valid = 0	out_en <= '0';
data_valid = 1	ram_wr_en <= c_GREEN; out_en <= '1';
Nezávislé výstupy	color_addr_ce <= '0'; color_addr_red <= color_addr - 1;
<i>Hodnota vstupov/Aktuálny stav</i>	st_b1
data_valid = 0	ram_wr_en <= "000";
data_valid = 1	ram_wr_en <= c_BLUE;
Nezávislé výstupy	color_addr_red <= color_addr; color_addr_ce <= '0'; out_en <= '0';
<i>Hodnota vstupov/Aktuálny stav</i>	st_dummy1
data_valid = 0	color_addr_ce <= '0'; out_en <= '0';
data_valid = 1	color_addr_ce <= '1'; out_en <= '1';
Nezávislé výstupy	color_addr_red <= color_addr; ram_wr_en <= "000";
<i>Hodnota vstupov/Aktuálny stav</i>	st_b2
data_valid = 0	out_en <= '0'; ram_wr_en <= "000";
data_valid = 1	out_en <= '1'; ram_wr_en <= c_BLUE;
Nezávislé výstupy	color_addr_ce <= '0'; color_addr_red <= color_addr - 1;



Obr. 20 Definícia prechodov medzi stavmi FSM *Control_FSM.vhd*

9.5.2 Prevodník RGB na Y

Teoretický princíp extrakcie jasovej zložky Y z komponent RGB je uvedený v kapitole 5. V tejto časti je vysvetlená reálna implementácia prevodníku RGB na Y do obvodu FPGA z obr. 16.



Obr. 21 Prevodník z RGB na Y (*rgb_to_y.vhd*)

Na obr. 21 je graficky znázornená entita prevodníku RGB na Y. V kóde VHDL je jeho názov *rgb_to_y.vhd*. Prevodník prevádza vstupné RGB komponenty v jednobajtovom formáte *R_i*, *G_i*, *B_i* na výstupnú jednobajtovú zložku *Y_o*. Obvod je z dôvodu dlhej kombinačnej cesty zo vstupov na výstup rozdelený klopnými obvodmi typu D mechanizmom pipeliningu (reťazenia). Nasledujúce riadky sú úryvkom behaviorálneho popisu tejto entity *rgb_to_y.vhd*.

```
--decimation factors for RGB
constant c_MUL : positive:= 3;
constant c_Rdiv : positive:= 10;
constant c_Gdiv : positive:= 5;
constant c_Bdiv : positive:= 10;

--internal auxiliary signals
signal s_Rm: unsigned(15 downto 0):=(others => '0');
signal s_Gm: unsigned(15 downto 0):=(others => '0');
signal s_Bm : unsigned(15 downto 0):=(others => '0');

begin

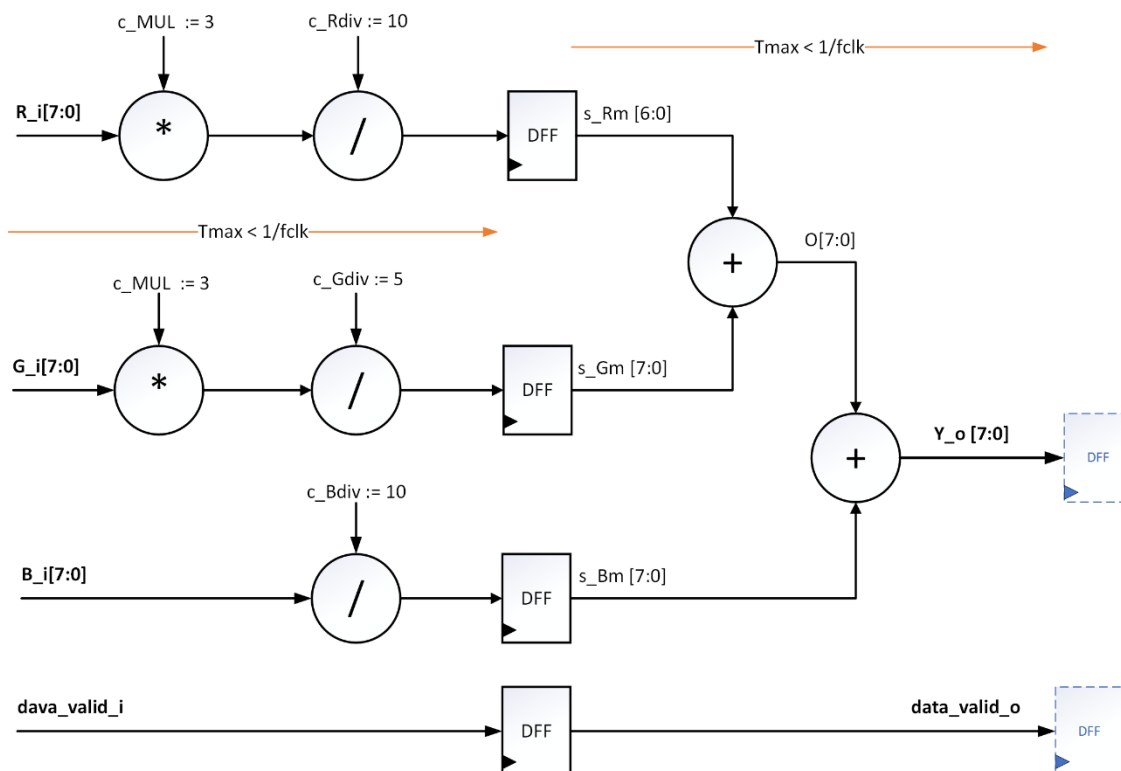
process(clk) begin
    if rising_edge(clk) then
        data_valid_o <= data_valid_i;
    end if;
end process;

pipeline:process(clk)
begin
    if rising_edge(clk) then
        s_Rm <= (c_MUL*unsigned(R_i))/c_Rdiv;
        s_Gm <= (c_MUL*unsigned(G_i))/c_Gdiv;
        s_Bm <= (x"00"&(unsigned(B_i)))/c_Bdiv;
    end if;
end process;
```

```
end process;
```

```
Y_o <= std_logic_vector(resize((s_Rm + s_Gm + s_Bm), 8));
```

Na obr. 22 je pre lepšiu prehľadnosť princípu dekodéru znázornený vyššie uvedený kód VHDL do ekvivalentného popisu na úrovni RTL v grafickej podobe. Kde DFF sú vyššie zmienené pipeliningové klopné obvody.



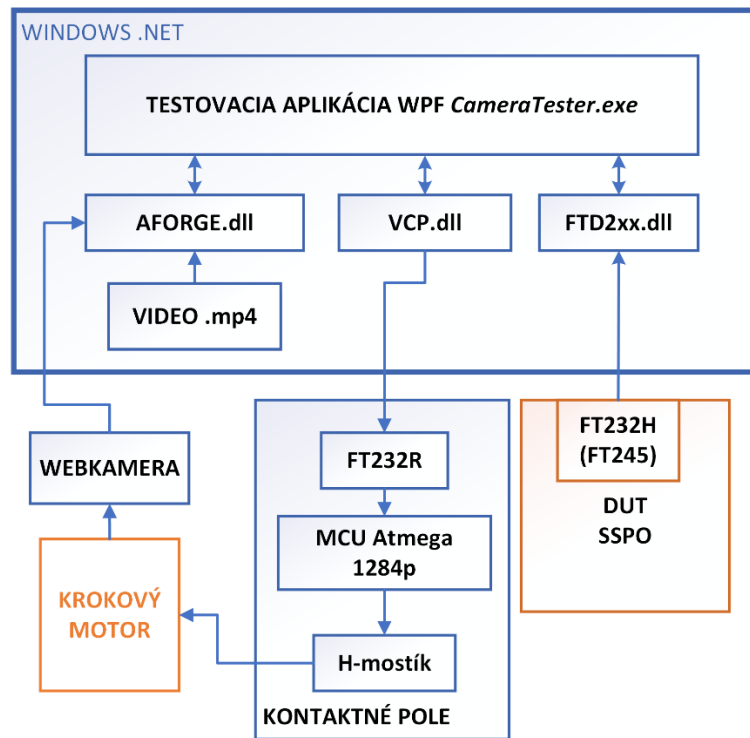
Obr. 22 Výsledný syntetizovaný návrh *rgb_to_y.vhd* na úrovni RTL

Po syntéze prevodníku *rgb_to_y.vhd* bola prevedená predbežná statická časová analýza. Jej výstupom sú hodnoty oneskorení všetkých kombinačných ciest v obvode. Najväčšie oneskorenia bolo zistené na ceste typu *clk_to_output* od *s_RM[2]* (výstup klopného obvodu) na výstup *Y_o[5]* s hodnotou 6,8 ns. Ostatné kombinačné cesty sú na veľmi podobnej úrovni oneskorenia v hodnotách do 6,5 ns. Z toho vyplýva, že kombinačná logika je v obvode správne vyvážená.

V prípade požiadavky na zvýšenie pracovnej frekvencie bude potrebné obvod ďalej “rozsekať” pridaním klopných obvodov medzi násobičku, deličku a dve sčítačky.

9.6 Návrh testovacej platformy algoritmov sledovania

Táto kapitola je venovaná vývinu jednoduchej platformy pre overenie behaviorálnej funkčnosti sledovacích algoritmov uvedených v teoretickej časti tejto práce. Platformu je možné rozdeliť do niekoľkých základných častí zobrazených na nasledujúcom obrázku.



Obr. 23 Grafické znázornenie testovacieho systému

Prvá časť aplikácie nachádzajúca sa v grafickom rozhraní CameraTester.exe umožňuje efektívne overiť funkčnosť algoritmov spracovávania obrazu a detekcie objektu pred ďalšou implementáciou do obvodu FPGA. V tejto časti je možné otestovať získanie rozdielových snímok popísaných v kapitole 6 a to buď z videosekvencie vo formáte .mp4 alebo priamo z webkamery, ktorá umožní otestovanie algoritmov v reálnom prostredí. Použitá kamera má rozlíšenie 640x360 px so snímkovacou frekvenciou okolo 25-30 Hz závislej od úrovne osvetlenia. Webkamera je upevnená na krokovom motore a je pripojená pomocou USB portu do PC. Vstupný obraz je v aplikácii spracovávaný v samostatnom vlákne, ktorého výstupom je pozícia pohybujúceho sa objektu. Táto pozícia je následne „zabalená“ do paketu, ktorý je posielaný cez USB port a prevodník FT232 do mikrokontroléra. Ako už bolo spomenuté na kontaktnom poli je pripojený mikrokontrolér ATmega1284, prevodník USB rozhrania na UART a H-mostík pre budenie krokového motora.

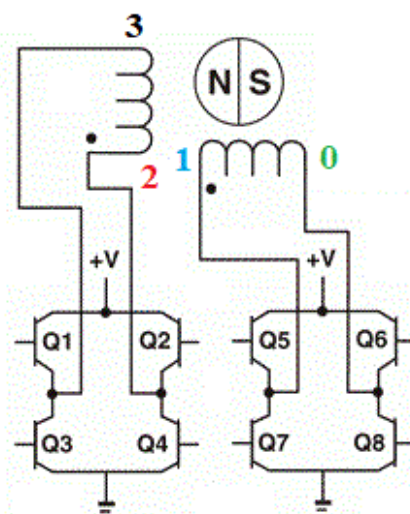
V mikrokontroléri je implementovaný Moorov stavový automat FSM (Finite State Machine), ktorého vstupom je dekodovaná správa z aplikácie s obsahom uhlového natočenia. Výstupom z automatu sú budiace impulzy pre H-mosť krokového motora. Bližšia špecifikácia riadenia krokového motora s odpovedajúcim riešením v podobe stavového automatu s diagramom prechodov a výstupov je uvedená v nasledujúcom texte.

Pozícia alebo aj rýchlosť ako už z názvu plyní sú ovládané impulzným charakterom riadiacich signálov. V našom prípade je použitá metóda Bipolar Half Stepping tzn. bipolárne budenie s posunom jedného polkroku. Počet krokov potrebných na vykonanie požadovanej uhlovej zmeny polohy v tejto metóde je vyjadrený ako

$$Ns = \frac{\alpha}{StepAngle} \times 2, \quad (11)$$

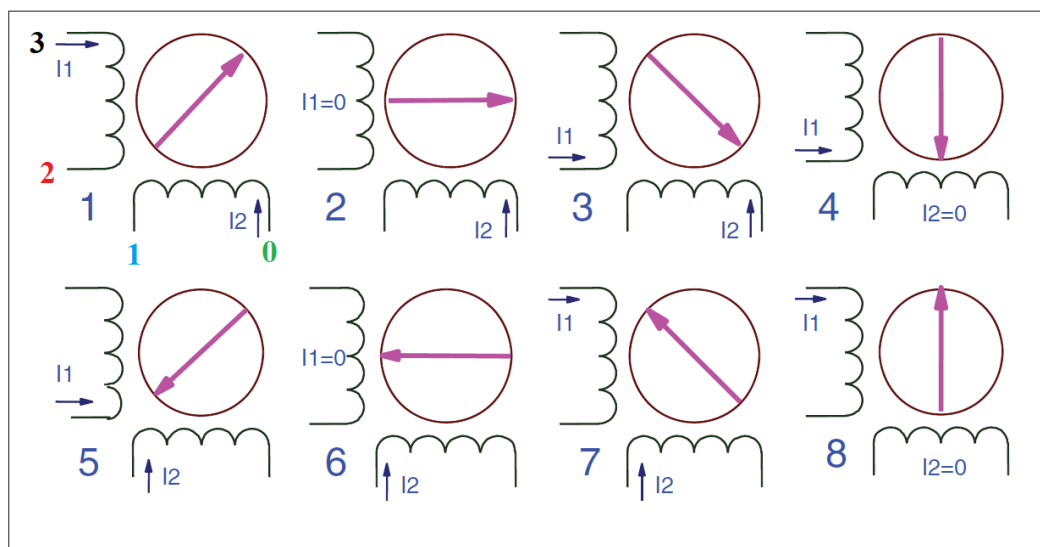
kde α je požadované pootočenie v stupňoch a StepAngle je známe uhlové pootočenie na jeden krok.

Zapojenie jednotlivých fáz vinutia motora a princíp komutácie sú znázornené na nasledujúcom obrázku. Vo väčšine prípadov je farebné označenie vývodov fáz ekvivalentné s číslovaním na obr. 24.



Obr. 24 Schematické zapojenie vinutia krokového motora do H-mosťka Zdroj:

https://www.st.com/content/ccc/resource/technical/document/application_note/57/c8/7c/c1/0d/91/46/89/CD00003774.pdf/files/CD00003774.pdf/jcr:content/translations/en.CD00003774.pdf



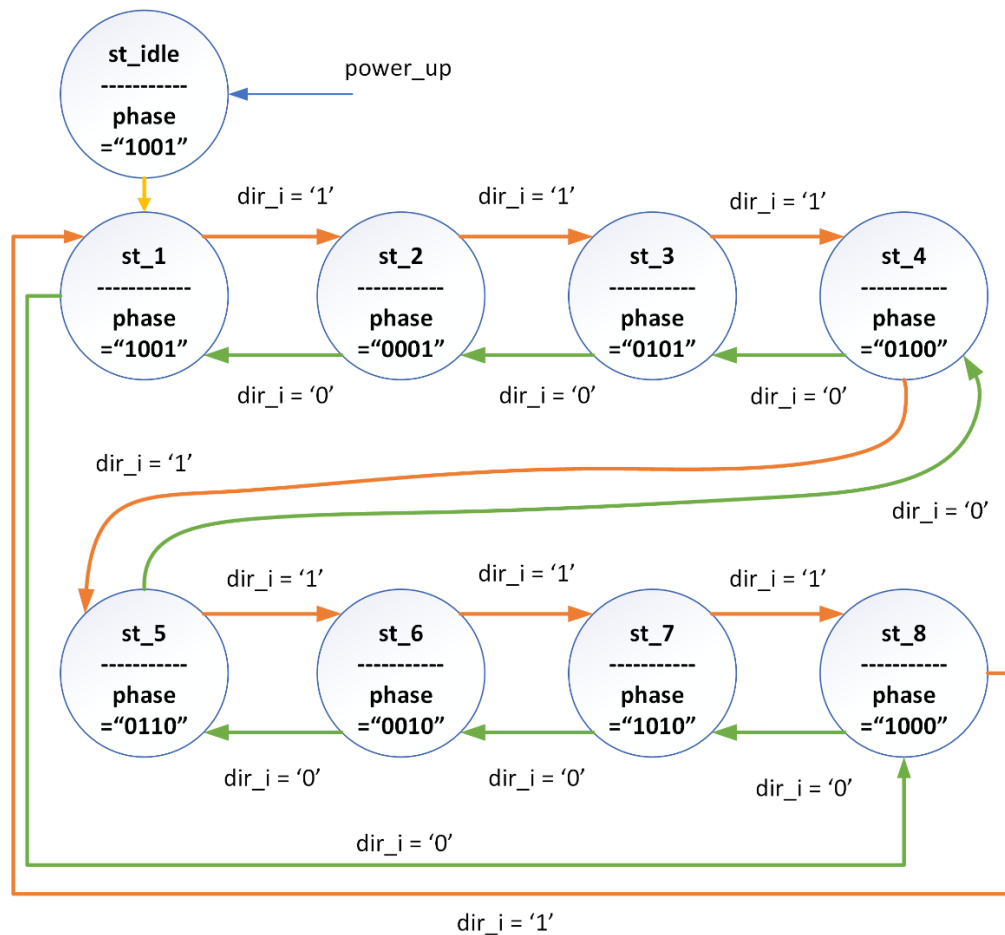
Obr. 25 Principiálny diagram bipolárneho pol krokového budenia krokového motora Zdroj: https://www.st.com/content/ccc/resource/technical/document/application_note/57/c8/7c/c1/0d/91/46/89/CD00003774.pdf/files/CD00003774.pdf/jcr:content/translations/en.CD00003774.pdf

Pre lepšie pochopenie je uvedený nasledujúci príklad jednej komutácie z pozície č.1 na pozíciu č.2 z obr. 25. V prvom kroku je na terminál č.1 pripojené napájacie napätie cez tranzistor Q1 a terminál č.2 je uzemnený cez tranzistor Q3. Ďalším krokom úplného odpojenia jednej z fáz je realizovaný posun. Nikdy nesmie nastať situácia simultánneho otvorenia tranzistorov na jednom termináli. Tranzistory musia byť určitú dobu komutácie uzatvorené aby nedošlo k skratu napájacieho napätia.

V prípade implementácie v SSPO je táto krízová situácia ošetrená interne v obvode LM 8548. Čip obsahuje interný oneskorovací obvod, ktorý uvažuje rýchlosti spínania tranzistorov a pridáva tzv. Dead Time. Počas tohto časového intervalu je terminál fázy pripojený na vysokú impedanciu tzn. Q1 aj Q3 sú uzatvorené na veľmi krátky čas v ráde mikrosekúnd. Ako už bolo spomenuté, realizáciu postupnosti komutácie fáz je výhodné riešiť stavovým automatom. Grafické znázornenie prechodov tohto FSM s výstupným vektorom Phase s dĺžkou 4 bity sa nachádza na nasledujúcom obrázku. Štvorica signálov z vektoru Phase je priamo pripojená na výstupný port mikrokontroléra alebo FPGA. Každý prechod medzi stavmi musí trvať minimálne 1 milisekundu pre zabezpečenie minimálneho výsledného momentu motora. V MCU je FSM realizovaný v podobe podprogramu, kde prechody medzi stavmi sú podmienené rozhodovacou premennou `dir_i` typu bool.

Realizácia FSM v obvode FPGA je v podobe sekvenčného logického obvodu, ktorého prechody sú podmienené príchodom nábežnej hrany a povoločovacieho signálu CE na klopné obvody typu D. Signál CE je generovaný komparátorom, ktorý porovnáva hodnotu čítača s preddefinovanou konštantou, ktorá určuje výslednú rýchlosť otáčania

motora. Na nasledujúcej strane je stavový diagram FSM s deklaračnou časťou entity `stepper_motor_controller.vhd`.



Obr. 26 Zobrazenie podmienených prechodov FSM s výstupom Phase

Treba podotknúť, že maximálny uhol pootočenia je limitovaný vstupným bitový rozsah signálu `angle_i` na 460°. Smer pootočenia je ovládaný signálom `dir_i`. Krokový motor môže byť odpojený od napájania vstupným signálom `shut_down_i`, ktorý ovláda výstupný multiplexor na signále `phase`.

Deklaračná časť entity `stepper_motor_controller.vhd`:

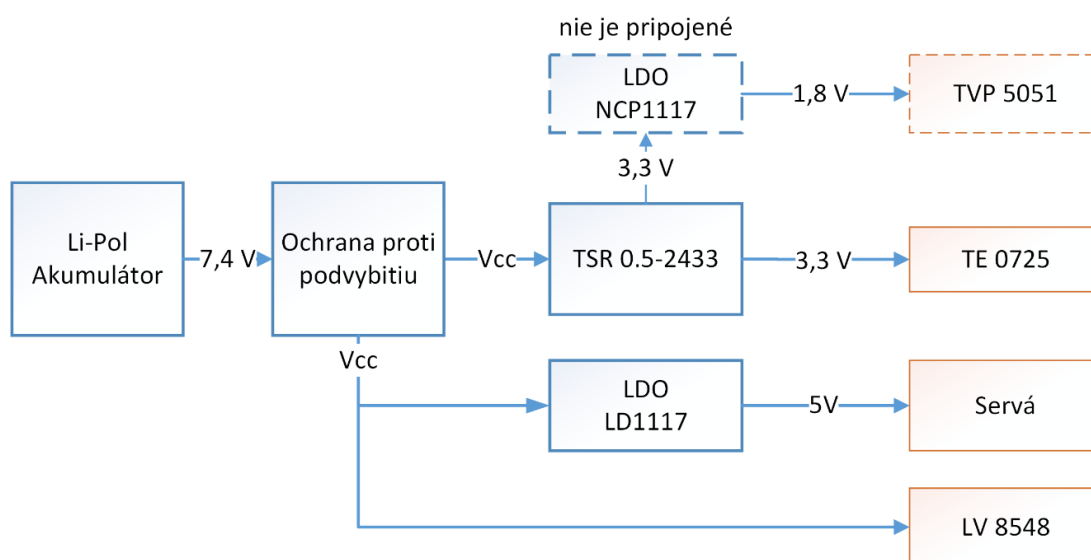
```

entity stepper_motor_controller is
    port (
        clk : in STD_LOGIC;
        rst : in STD_LOGIC;
        angle_i : in UNSIGNED(8 downto 0); --512 half-steps resolution
        step_done_i : in STD_LOGIC;
        dv_i : in STD_LOGIC;
        dir_i : in STD_LOGIC;
        shut_down_i : in STD_LOGIC;
        phase_o : out STD_LOGIC_VECTOR (3 downto 0));
end stepper_motor_controller;

```

9.7 Distribúcia napájania s ochranou batérie

Systém je napájaný z dvojčlánkovej Li-Pol batérie. Vzhľadom na to, že RJ neobsahuje žiadne analógové obvody citlivé na šum, nie je výstup DC/DC meniča filtrovaný cez LDO ale iba paralelnou kombináciou elektrolytického a multivrstvého MLCC kondenzátora, ktoré spoľahlivo pokrývajú prúdové špičky a odfiltrujú zvlnené napätie na prípustnú hodnotu. Na obr. 27 je grafické znázornenie distribúcie napájacieho napätia na RJ. Na batériu je pripojený jednoduchý obvod UVP (Under Voltage Protection) sledujúci jej napätie, ktorý v prípade jeho poklesu odpojí napájacie napätie pre ďalšie časti RJ. Za týmto obvodom je DC/DC (direct current – direct current) menič TSR 0.5-2433 od firmy Traco Power, ktorý zabezpečí zníženie napätia batérie s vysokou účinnosťou až 95 %. Menič od výrobcu Traco Power bol vybraný spomedzi konkurentov z dôvodu extrémne malých rozmerov s nízkou hmotnosťou, ale zároveň má vynikajúce elektrické parametre, ktoré sú popísané ďalej. Pre napájanie serv je použitý lineárny stabilizátor ST LD1117 v puzdre DPAK s výstupným napätím 5V a maximálnym trvalým prúdom 500 mA. [15]



Obr. 27 Bloková schéma distribúcie napájania na RJ

Tab. 3 Popis prúdového odberu a spôsob napájania jednotlivých zariadení [16] [15]

Napájacie napätie	Zariadenie	Maximálny odber [mA]
3,3	TE 0725	~400
1,8	TVP 5150	60
5	HS-55 servá	1000
7,4	krokový motor SY35ST26-0284A	280
Súčet		< 1740 mA

Na obr. 28 je znázornená ochrana celého systému proti prepólovaniu napájacieho napätia Shottkyho diódou (NXP PMEG3010ER) s nízkym prahovým napätím. V prípade reverzácie napájania so zemou nebude dióda otvorená, obvodom nebude tiecť ideálne žiadny prúd. Nevýhodou tohto riešenia je permanentný úbytok napätia, ktorý sa pohybuje okolo 223 mV pri prúdovom odbere 220 mA. Z toho vyplýva aj stratový výkon 49 mW, ktorý diódu zahrieva. Pre výpočet maximálneho stratového výkonu s použitím puzdra SOD-123W pri teplote okolia 50 °C a maximálnej teplote PN prechodu 80 °C platí

$$P_{d(max)} = \frac{T_j - T_A}{R_{th(j-a)}} = \frac{353,15K - 323,15K}{220 K/W} = 136mW > 49 mW, \quad (12)$$

kde T_j [K] je maximálna prípustná teplota PN prechodu. T_a [K] je teplota okolia, a $R_{th(j-a)}$ je tepelný odpor medzi PN prechodom a okolitým prostredím pri minimálnej veľkosti spájkovacích plošiek. Z vyššie uvedeného vzťahu vyplýva, že teplota diódy nepresiahne 80 °C. [17]

Maximálny odhadovaný stratový výkon na LDO IC9 je

$$\begin{aligned} P_{d,IC9} &= (U_{in} - U_{out}) \times I_{(+5V)} = (7,4V - 5V) \times 500 mA \\ &= 1,2 W \end{aligned} \quad (13)$$

kde $I_{(+5V)}$ je maximálny odhadovaný prúd pre vetvu napájania + 5 V.

Tepelná limitácia stratového výkonu na LDO IC9, s typovým označením DPAK (SINGLE GAUGE) CASE 369S ISSUE E je

$$\begin{aligned} P_{d(max) IC9} &= \frac{T_j - T_A}{R_{th(j-a)}} = \frac{120^\circ C - 25^\circ C}{67^\circ \frac{C}{W}} \times 0,5 = 1,4 W \\ &> (P_{d,IC9}), \end{aligned} \quad (14)$$

kde T_j [°C] je maximálna prípustná teplota PN prechodu, T_a [°C] je teplota okolia. $R_{th(j-a)}$ je tepelný odpor medzi PN prechodom a okolitým prostredím pri minimálnej veľkosti spájkovacích plošiek. V nasledujúcej tabuľke sú uvedené dôležité parametre DC/DC meniča. [17] [16]

Tab. 4 Parametre DC/DC konvertora Traco Power TSR 0.5-2433 [15]

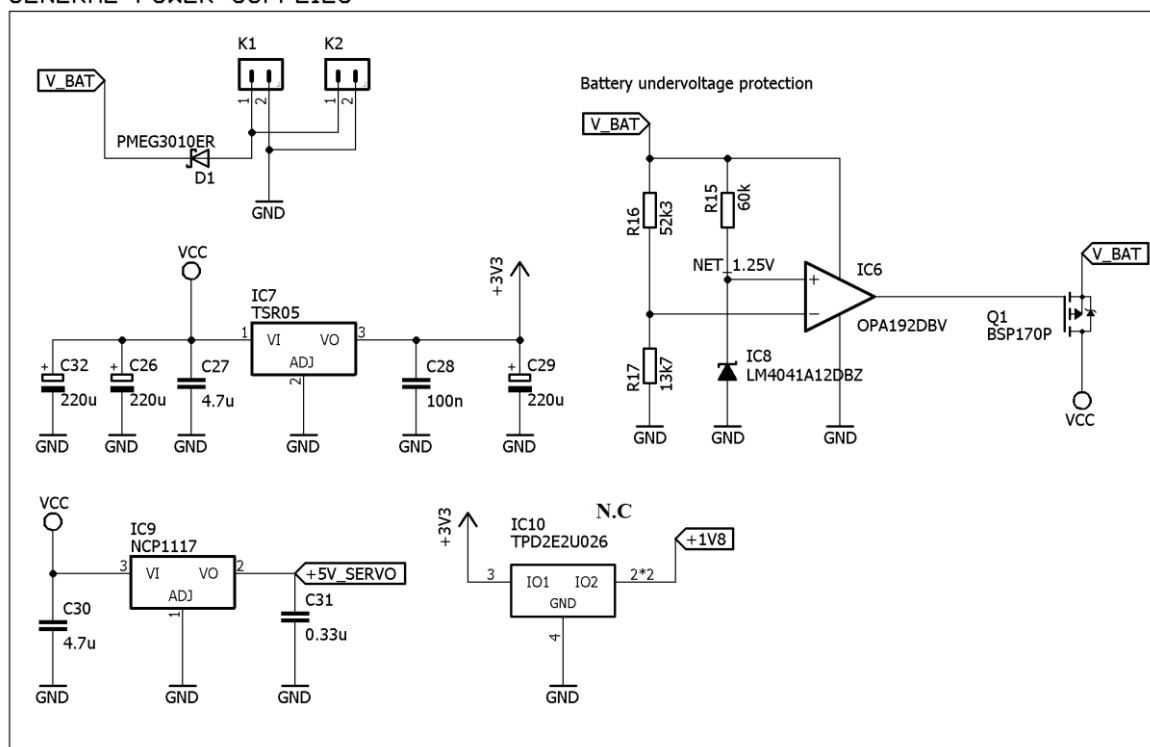
Rozsah vstupného napätia	4,75 – 32 V
Hodnota výstupného napätia	3,3 V
Maximálne zvlnenie na výstupe	30 mV p-p
Presnosť výstupného napätia	± 3 %
Spínacia frekvencia	330 kHz ± 50kHz (PWM)
Ochrana proti skratu	áno

V tejto časti bude podrobnejšie vysvetlená oblasť ochrany proti podvybitiu – Battery undervoltage protection. Na obr. 28 je znázornené schematické zapojenie napájania celého systému. Invertujúce zapojenie komparátora IC6 porovnáva zmenšené napájacie napätie deličom R16-R17 s presnou referenčnou hodnotou napätia vytvorenou napäťovo stabilnou referenciou LM 4041. Odpor R15 bol zvolený tak, aby prúd pretekajúci referenciou bol v pracovnom pásme preto

$$R15 = \frac{(U_{BATmin} - U_{ref})}{I_{ref}} = \frac{(6V - 1,25V)}{65 \mu A} \cong 60 k\Omega . \quad (15)$$

Deliacim pomerom odporového deliča nastavíme minimálne napätie batérie, resp. napäťový bod preklopenia komparátora uzatvorenia tranzistora. Systém je napájaný z dvojčlánkovej Li-Pol batérie, ktorej kritická hodnota napätia na jeden článok sú 3 V. Kritické napätie pri ktorom musí byť systém od batérie odpojený je 6 voltov. Z dôvodu minimalizácie prúdovej spotreby hľadáme odpory v rádoch kiloohmoch s deliacim účinkom vstupného napätia číslom päť. Prahové napätie s chybou menej ako 5% vychádza najlepšie s odporom R16 s hodnotou 52k3 a odporom R17 s hodnotou 13k7 .

GENERAL POWER SUPPLIES



Obr. 28 Obr. Schematické zapojenie distribúcie napájacích napätí na DPS RJ

9.8 Implementácia detekcie pohybujúceho sa objektu v .NET aplikácii

Overenie teoretických predpokladov extrakcie statického pozadia z kapitoly 6 je výhodné otestovať ešte pred samotnou implementáciou do obvodov FPGA. Akákoľvek zmena architektúry na úrovni RTL v FPGA je časovo náročnejšia a neefektívna v porovnaní so zmenou programového kódu v aplikácii pre operačný systém Windows. V tejto fáze bol zároveň aj vylepšený algoritmus generovania prahovej hodnoty dif_{THR} z kapitoly 6, ktorá definuje rozdiel medzi pohybom a statickým pozadím. Táto hodnota sa adaptívne mení v závislosti na percentuálnom zastúpení celkového pohybu v obrazových blokoch o rozmeroch 8x8 pixelov.

Kompletný upravený postup získania pohybu v obraze v aplikácii je možné definovať nasledovne:

1. Vytvorenie absolútnych diferencií jedno-bajtových obrazových bodov medzi starým a novým snímkom.
2. Rozdelenie obrazu na bloky 8x8 pixelov.
3. Vypočítanie priemerného rozdielu v jednotlivých blokoch, ktoré definuje priemerný pohyb v tejto časti obrazu.
4. Porovnanie vypočítaného priemerného rozdielu s pôvodným obrazom v rámci blokov 8x8 pixelov.

Po získaní pohybu v obraze už nie je žiadny problém dopočítať súradnice objektu podľa vzťahu č.8 v kapitole 15 a dokresľovať ich v reálnom čase do aplikácie spolu s grafickým ohraničením.



Obr. 29 Demonštrácia funkčnosti lokalizácie pohybujúceho sa objektu v statickom obraze z videosekvencie

9.9 Filtrácia malých objektov v binárnom obraze

Táto kapitola sa venuje implementácii filtrácie binárnych rozdielových snímok popísaných v kapitole 7. Nasledujúci text popisuje komponentu `filtration_unit.vhd`, ktorá zapisuje a číta dáta z blokových pamätí BRAM.

Deklaračná časť entity `filtration_unit.vhd`:

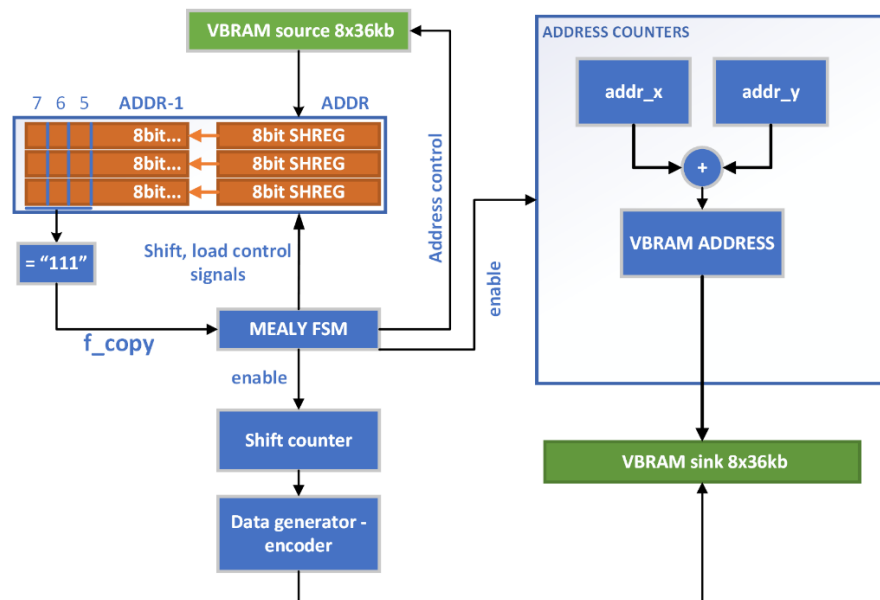
```
entity filtration_unit is
    generic(C_WINDOW_SIZE: natural :=3;
           C_BRAM_PW: natural :=8;
           H_RES: natural := 800);
    port (   clk : in std_logic;
            rst : in std_logic;
            ready_i : in std_logic;
            busy_o : out std_logic;
            done_o : out std_logic;
            ----- VBRAM -----
            bram_addr_o : out std_logic_vector(11 downto 0);
            -- SINK VBRAM MEMORY --
            bram_sink_wr_en_o : out std_logic;
            bram_sink_data_o : out std_logic_vector(C_BRAM_PW-1 downto 0);
            bram_sink_wr_done_i : in std_logic;
            -- SOURCE VBRAM MEMORY --
            bram_source_data_i : in std_logic_vector(C_BRAM_PW-1 downto 0)
    );
end filtration_unit;
```

Základnými generickými parametrami komponenty `filtration_unit.vhd` sú

- `C_WINDOW_SIZE` – veľkosť filtračného okna,
- `C_BRAM_PW` – šírka dátovej zbernice video blokových pamätí VBRAM,
- `H_RES` – horizontálne rozlíšenie vstupného obrazu.
- Základnými vstupnými a výstupnými signálmi sú
- `clk` – hodiny,
- `rst` – synchrónny reset,
- `ready_i` – startuje začiatok filtrácie resp. reprezentuje pripravenosť predchádzajúceho bloky, že vstupná pamäť `bram_source_data_i` je naplnená novými dátami,
- `busy_o` – informuje ďalšie entity o zaneprázdnenosti,
- `done_o` –informuje o skončení filtrácie celého obsahu jednej BRAM pamäte,
- `bram_addr_o` – adresa pre zápis a čítanie z/do pamätí BRAM,
- `bram_sink_wr_en_o` – povoloovací signál pre zápis do pamäte VBRAM_sink8x36kb,
- `bram_sink_data_o` – dáta pre zápis do VBRAM_sink8x36kb (výsledná osmica odfiltrovaných pixelov),
- `bram_sink_wr_done_i` – informuje o úspešnom zapísaní do blokovej pamäte VBRAM_sink8x36kb,

- bram_source_data_i – zdrojové dáta z blokovej pamäte VBRAM_source8x36kb.

Principiálne je filtrácia obrazu vysvetlená v kapitole 7. Komponenta bola navrhnutá tak aby bol vstupný obraz rozdelený do 15-tich blokov po 40-tich riadkoch, pričom každý blok je filtrovaný nezávisle jednou komponentou filtration_unit.vhd. Tento spôsob zaručí optimálne využitie blokových pamätí a najmä urýchli spracovanie obrazu. Na obr. 30 je načrtnutá interná architektúra filtračnej entity. Základným jadrom celého systému je Mealyho stavový automat, riadiaci vstupných dát z blokovej pamäte VBRAMsource8x36kb do trojice 16-bitových posuvných registrov. Dáta sú posúvané dovtedy, kým posledná trojica bitov nie je v každom posuvnom registri v log.1. Ak nastane situácia, že je posledná trojica bitov v každom posuvnom registri nastavená do log.1 (testovacie okno z kapitoly 7 je zaplnené) je týchto 9 bitov prekopírovaných do VBRAMsink8x36kb na príslušnú adresu. Obraz je v dedikovaných blokových pamätiach uložený po bajtoch z dôvodu rýchlejšieho spracovania a minimalizácie adresných dekodérov. Globálne je obraz uložený riadok za riadkom, pričom na adrese nula začína nultý horný riadok obrazu. Treba si uvedomiť, že môže nastať situácia kedy, bude kontrolné okno zaplnené z dvoch trojíc adries. Tieto medzi adresné prechody sú vyriešené v stavovom automate. Okrem FSM sú v návrhu použité štyri čítače na zapamätanie kroku posunu v rámci posuvného registra, dva adresné čítače addr_x, addr_y a horizontálny čítač cnt_cr polohy vo filtračnom okne.

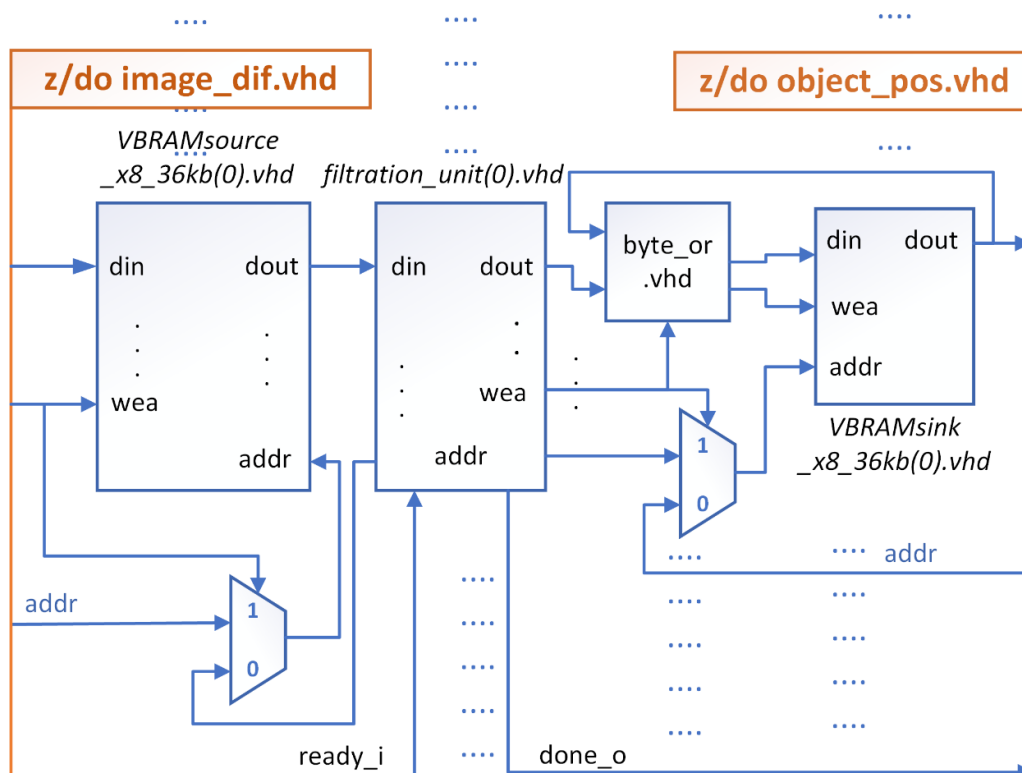


Obr. 30 Bloková schéma *filtration_unit.vhd*

Principiálne zapojenie filtračnej jednotky do SSPO bez znázornených kontrolných a pomocných signálov je na obr. 31. Jednotka filtration_unit.vhd je po kompletom zaplnení VBRAMsource_x8_36kb novými dátami informovaná signálom ready_i. Po

asertovaní tohto signálu do log.1 prejde FSM (jednotlivé prechody sú znázornené na ďalšej strane na obr. 32) zo stavu `st_idle` a začne čítať vstupnú pamäť `VBRAMsource_x8_36kb`.

Zápis bajtov do pamäte `VBRAMsink_x8_36kb` na rovnakú adresu musí byť logickým súčtom nových dát so starými aby nedochádzalo k ich mazaniu. Na schéme z obr. 31 je vidieť, že zápis do oboch pamätí má prednosť pred čítaním. O kompletnom zaplnení pamäte `VBRAMsink_x8_36kb` novými dátami je entita `object_pos.vhd` informovaná signálom `done_o`.



Obr. 31 Približné schematické začlenenie jednotky filtrácie `filtration_unit.vhd` do celého systému

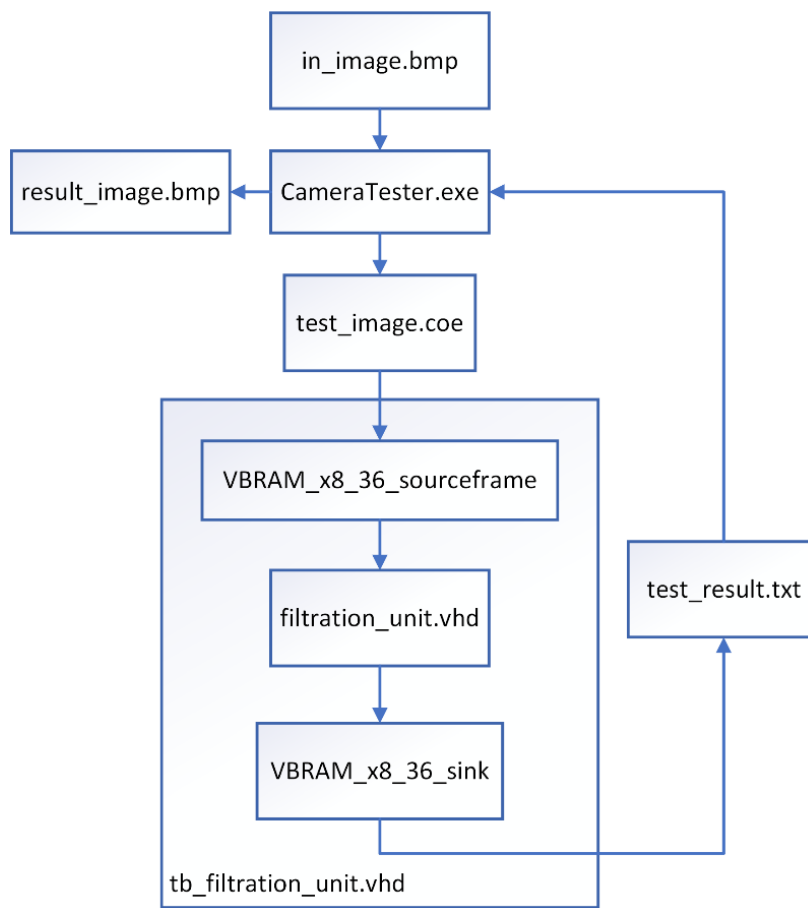
Tabuľka výstupov nie je v texte z dôvodu výpovednej a prehľadovej stránky uvedená, ale jej kompletný HD návrh je dostupný v elektronickej prílohe tejto práce. Približný popis jednotlivých stavov a ich výstupov je nasledovný

- `st_idle` – čítače a posuvné registre sú zablokované, výstupné signály sú v log. 0,
- `st_loadMSB` – čítač načítania vstupných dát z pamäti je povolený a je resetovaný ak nadobúda hodnotu 3, dáta sú z blokovej pamäti načítané do MSB posuvných registrov cez multiplexor,
- `st_loadLSB` - čítač načítania vstupných dát z pamäti je povolený a je resetovaný ak nadobúda hodnotu 3, dáta sú z blokovej pamäti načítané do LSB posuvných registrov cez multiplexor,
- `st_addr_x_plus` – posuvné registre a čítače okrem adresy `x` sú zablokované,
- `st_shift` – dáta sú posúvané doľava smerom od LSB po MSB bit,
- `st_last` – bude načítaných posledných 8 bitov z adresy `VBRAMsource`,
- `st_addr_x_clr` – pre prechod na nový riadok v obraze je nevyhnutné vynulovať adresný čítač `addr_x`, a zároveň inkrementovať ukazateľ resp. čítač `addr_y` a vyresetovať ukazateľ pozície v rámci jedného bajtu,
- `st_test` – čítače a posuvné registre sú zablokované aby kombinačná logika prechodov mohla zaznamenať výstup z komparátora `f_copy`,
- `st_copyLSB` – asertuje sa povolovací signál pre zápis do `VBRAMsource`,
- `st_copyMSB` - asertuje sa povolovací signál pre zápis do `VBRAMsource`,
- `st_done` – asertuje informačný signál `done_o` do log.1 po dobu jednej periódy `clk`.

9.9.1 Verifikácia entity `filtration_unit.vhd`

Pre otestovanie behaviorálnej funkčnosti bol vytvorený jednoduchý testovací proces s využitím aplikácie `CameraTester.exe` popísanej v kapitole 9.6

V aplikácii je po stlačení tlačidla s názvom `Generate RAM COE FILE` zavolaná metóda, ktorá načíta bitmapový vstupný zašumený obrázok (vytvorený napríklad v obyčajnom skicári), z ktorého vygeneruje textový súbor koeficientov `test_image.coe` pre blokovú pamäť `VBRAM_x8_36kb_sourceframe`. Pri spustení behaviorálnej simulácie v IDE Vivado kompilátor načíta tento súbor do pamäte. Ďalej je vygenerovaný signál `ready_i` pre naštartovanie samotnej filtrácie. Asertovaním signálu `done_o` je obsah pamäte `VBRAM_x8_36kb` čítaný a zapisovaný do súboru `test_result.txt`. Spätnou rekonštrukciou súboru je v aplikácii `CameraTester.exe` vytvorený výsledný odfiltrovaný obraz. Grafické porovnanie vstupného a výstupného obrazu je znázornené na obr. 34 nasledujúcej strany.



Obr. 33 Blokové znázornenie testovacieho jednotky filtration_unit.vhd



Obr. 34 Porovnanie vstupného zašumeného obrázka in_image.bmp (nad červenou čiarou) s výsledným odfiltrovaným obrázkom result_image.bmp

Na obr. 34 je vidieť, že blok filtrácie funguje správne a všetky rušivé elementy o veľkosti menšej ako 3x3 pixely sú z obrazu odstránené.

Predbežné využitie zdrojov FPGA po syntéze `filtration_unit.vhd` je znázornené v nasledujúcej tabuľke.

Tab. 5 Štatistika využitia zdrojov v FPGA po syntéze

	Využitých [-]	Celkový počet dostupných [-]	Percentuálne využitie [%]
FF	122	41600	1,34
LUT	278	20800	0,29

Ďalším dôležitým parametrom je časový interval za ktorý bude filtračná jednotka schopná odfiltrovať celý obsah vstupnej blokovej pamäte obsahujúci 40 riadkov s dĺžkou 800 bodov (na konci riadka je pridaný jeden dummy pixel, reálna dĺžka je 799 obrazových bodov). Z behaviorálnej simulácie bola pri pracovnej frekvencii 100 MHz odčítaná hodnota tohto intervalu 1 milisekunda. Tento výsledok je vynikajúci pretože sa zhoduje s predbežne stanoveným odhadom počas návrhu. Filtrácia celého obrazu v FPGA je aj pri tak malej pracovnej frekvencii stále rýchlejšia ako jeden proces bežiaci v testovacej aplikácii `CameraTester.exe`.

9.10 Generátor PWM impulzov pre riadenie servomotorov

Táto kapitola sa venuje implementácii PWM rozhrania pre ovládanie servomotorov s rozlíšením 0,9°/LSB.

Deklaratívna časť entity:

```
entity servo_controller is
    generic(F_CLK : natural:= 100000000; F_PWM: natural:=50);
    Port ( clk : in STD_LOGIC;
          rst : in STD_LOGIC;
          angle_i : in signed(11 downto 0);
          pwm_o : out STD_LOGIC);
end servo_controller;
```

Základnými generickými parametrami komponenty `servo_controller.vhd` sú

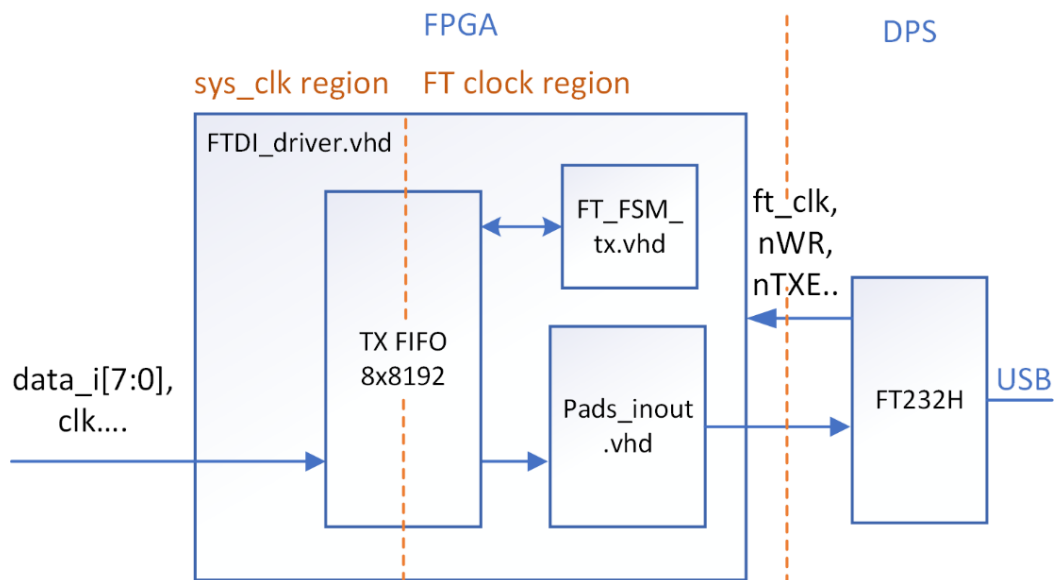
- F_CPU – taktovacia frekvencia signálu `clk`
- F_PWM– frekvencia pulznej modulácie serva.
- Základnými vstupnými a výstupnými signálmi sú
- `clk` – globálny hodinový signál,
- `rst` – synchronný reset,
- `angle_i` – vstupný uhol v rozsahu $\pm 460^\circ$,
- `pwm_o` - výstupný PWM signál.

Principiálne sa v komponente nachádzajú dva čítače, násobička vstupného uhla angle_i s prevodnou konštantou K , sčítacia a komparátor. Prvý čítač p_slow_20MHz generuje povoľovací signál pre druhý čítač p_pwm a tým určuje výsledné rozlíšenie PWM, ktoré je 50 nanosekúnd. Priebeh signálu pwm_o v čase je generovaný multiplexorom, ktorý je riadený komparátorom porovnávajúcim hodnotu čítača p_pwm s neutrálnou hodnotou C_CNT_NEUTRAL . Výstupný signál pwm_o nadobúda hodnotu $\log.1$ iba ak je hodnota čítača p_pwm menšia ako súčet konštanty C_CNT_NEUTRAL a vstupnej hodnoty angle_i vynásobená konštantou K .

9.11 Vysokorýchlostný verifikačný kanál FT245 s generátorom správy

Táto kapitola je venovaná realizácii vysokorýchlostného postranného verifikačného rozhrania medzi aplikáciou CameraTester a obvodom FPGA. Jeho primárnou úlohou je zjednodušiť vývoj zariadenia a preposielať obrazové dáta do testovacej aplikácie v reálnom čase. Maximálna teoretická bitová priepustnosť je limitovaná rýchlosťou USB 480 Mbit/sekundu.

Vysokorýchlostné spojenie medzi PC a FPGA je realizované obvodom FT232H, ktorý s FPGA komunikuje cez paralelné FIFO rozhranie FT245. Zatiaľ je v FPGA implementovaná len vysielať časť tzn. dáta smerujú iba z FPGA do PC. Principiálne zapojenie tejto entity do systému SSPO je znázornené na obr. 35.



Obr. 35 Zapojenie najnižšej komunikačnej vrstvy (ovládača) medzi FT232 a SSPO

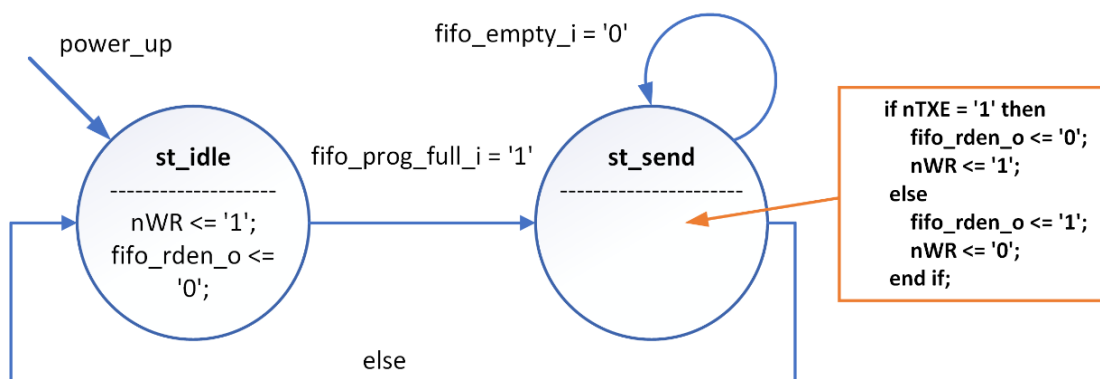
Deklaratívna časť entity:

```
entity FTDI_driver is
  Port ( clk : in STD_LOGIC;
        fclk : in STD_LOGIC;
        data_i : in STD_LOGIC_VECTOR (7 downto 0);
        data_valid_i : in STD_LOGIC;
        nWR : out STD_LOGIC;
        nTXE : in STD_LOGIC;
        rst : in STD_LOGIC;
        data_o : out STD_LOGIC_VECTOR (7 downto 0);
        status_o : out std_logic
  );
end FTDI_driver;
```

Základnými vstupnými a výstupnými signálmi entity FTDI_driver.vhd sú

- clk – globálny hodinový signál,
- rst – synchronný reset,
- fclk – vstupný hodinový signál pre vyčítavanie dát vyrovnávacej pamäte do FT232H,
- data_i – vstupné dáta,
- data_valid_i – zapisovací signál do vyrovnávacej pamäte,
- nWR – zapisovací signál do vyrovnávacej pamäte v FT232H (aktívny v log.0),
- nTXE – vstupný signál, ak je v log.0 FT 232 je pripravený vyčítavať dáta z TX FIFO v FPGA.

Ako bolo spomenuté v úvode, ovládač *FTDI_driver.vhd* je zložený z vysielacieho FIFa, ktorého zápis a čítanie sú riadené jednoduchým stavovým automatom FT_FSM_tx.vhd, ktorý má dva pracovné stavy *st_idle* a *st_send*. V stave *st_idle* čaká na zaplnenie vysielacieho FIFa dátami a v stave *st_send* povoľuje čítanie FIFa podľa vstupného signálu *nTXE*.



Obr. 36 Stavový diagram Mealyho vysielacieho FSM pre FT232H

Ďalšia časť textu je venovaná vyššej komunikačnej vrstve, ktorá zabaľuje vstupné dáta pre FTDI_driver.vhd do paketu, ktorý je dekodovaný v aplikácii CameraTester. Aby bolo možné paket dekodovať, musí začínať a končiť tzv. identifikačným reťazcom. Najjednoduchším reťazcom je kombinácia dvoch po sebe idúcich znakov označovaných ako ESP_SOP0 (Escape character for Start of Packet 0) a ESP_SOP1 (Escape character for Start of Packet 1). Ďalej je dôležité aby v obsahu paketu nenastala postupnosť týchto ESP znakov, pretože by prijímacia časť vyhodnotila tento falošný stav ako koniec alebo začiatok nového paketu. Z tohto dôvodu pokiaľ správa obsahuje znak ESP_SOP0, ESP_EOP0 (vhodné zvoliť numericky rovnakú hodnotu) je za tento znak priradený ďalší znak ESP_spare, ktorý zabezpečí, že správa nikdy nebude obsahovať falošnú postupnosť označujúcu začiatok alebo koniec paketu.

Tab. 6 Štatistika využitia zdrojov v FPGA po syntéze

	Využitých [-]	Celkový počet dostupných [-]	Percentuálne využitie [%]
FF	11	41600	0,03
LUT	2	20800	0,01
BRAM	1	50	2

Deklaratívna časť entity:

```
entity esp_char_gen is
    generic(DATA_LEN : natural :=1599;
            ESP_SOP0 : std_logic_vector(7 downto 0):=x"ff";
            ESP_SOP1 : std_logic_vector(7 downto 0):=x"80";
            ESP_EOP0 : std_logic_vector(7 downto 0):=x"ff";
            ESP_spare : std_logic_vector(7 downto 0):=x"FE";
            ESP_EOP1 : std_logic_vector(7 downto 0):=x"90");
    Port ( clk : in STD_LOGIC;
          rst : in std_logic;
          start_i : in STD_LOGIC;
          data_i : in STD_LOGIC_VECTOR (7 downto 0);
          data_o : out STD_LOGIC_VECTOR (7 downto 0);
          data_valid_i : in STD_LOGIC;
          data_valid_o : out STD_LOGIC);
end esp_char_gen;
```

Základnými generickými parametrami komponenty esp_char_gen.vhd sú

- DATA_LEN – dĺžka posielanej správy,
- ESP_SOP0 – Espace Character for Start of Packet 0 – identifikačný znak začiatku správy č.0,
- ESP_SOP1 – Espace Character for Start of Packet 1 – identifikačný znak začiatku správy č.1,
- ESP_EOP0 – Espace Character for Start of Packet 0 – identifikačný znak konca správy č.0,

- ESP_EOP1 – Espace Character for Start of Packet 10– identifikačný znak konca správy č.0,
- ESP_spare – Espace Character for Start of Packet 0 – špeciálny znak.

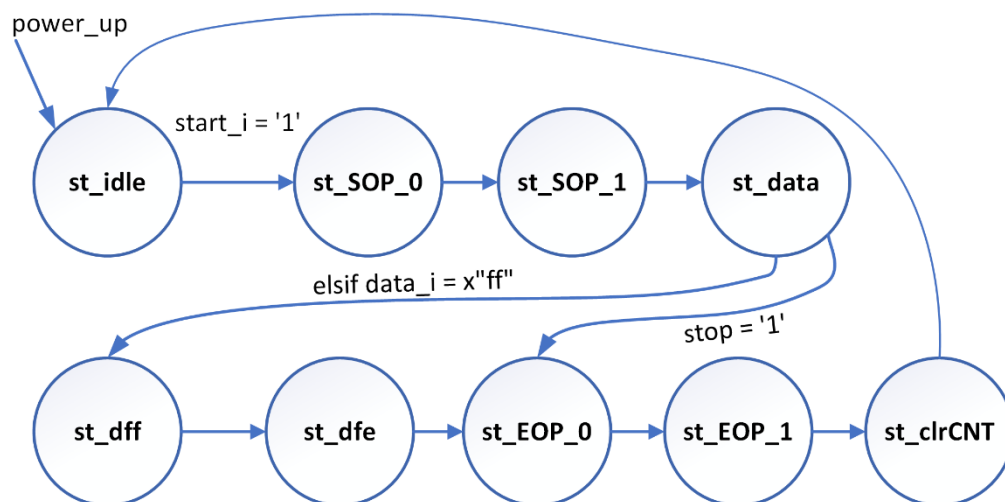
Vstupmi a výstupmi sú

- clk – globálny hodinový signál,
- rst – synchronný reset,
- data_i – vstupný dátový stream,
- data_o – výstupný dátový stream,
- data_valid_i - validačný vstupný signál,
- data_valid_o – výstupný validačný signál.



Obr. 37 Štruktúra paketu

Generácia paketu z obr. 37 na predchádzajúcej strane je implementovaná v Mealyho stavovom automate ESP_CHAR_GEN.vhd. V prípade, že bude signál start_i v log.1 prejde automat postupne do stavu st_SOP_0 v ktorom vygeneruje výstupný validačný signál valid_o a multiplexorom nastaví dátovú zbernicu data_o na hodnotu ESP_SOP0. Doplnenie ostatných znakov je takisto vyriešené týmto spôsobom prostredníctvom multiplexora. Ošetrenie generácie falošných postupností ESP znakov je riešené porovnávaním vstupných dát komparátorom s hodnotami ESP_SOP0 a ESP_EOP0. V prípade, že komparátor nájde zhodu prejde automat do stavu st_dff a st_dfe v ktorom znovu cez ten istý výstupný multiplexor zmení výstupnú hodnotu data_o na hodnotu ESP_spare a asertuje signál pre zápis valid_o.



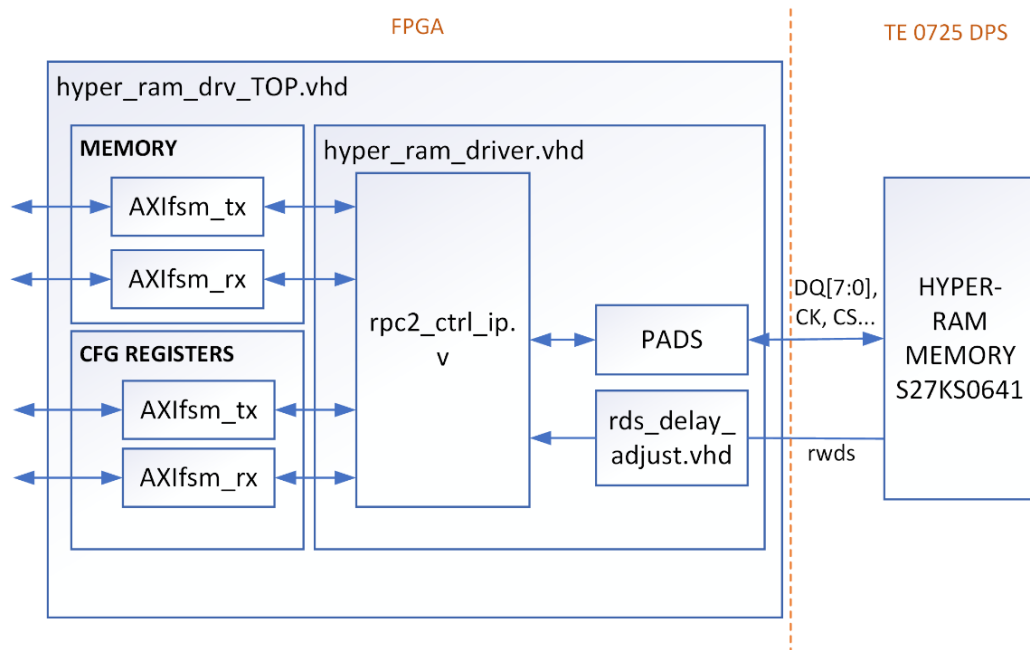
Obr. 38 Diagram prechodov generátora ESP znakov

Tab. 7 Štatistika využitia zdrojov v FPGA po syntéze

	Využitých [-]	Celkový počet dostupných [-]	Percentuálne využitie [%]
FF	20	41600	0,05
LUT	30	20800	0,14

9.12 Implementácia jednotky pre separáciu statického pozadia

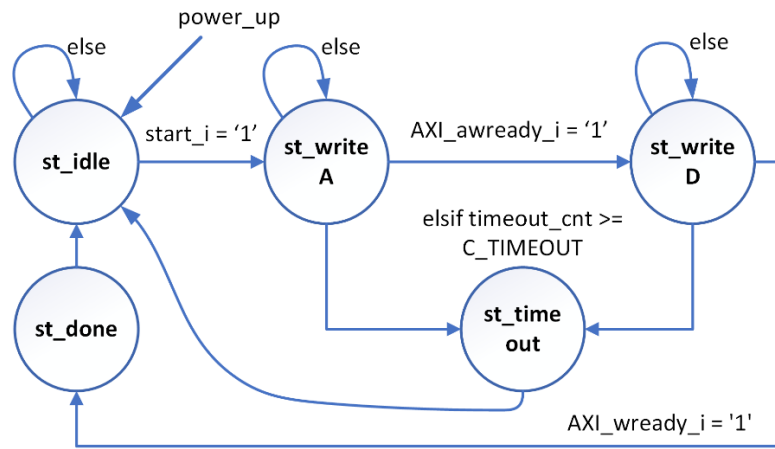
Táto časť projektu je v štádiu vývoja. Zatiaľ bol navrhnutý ovládač `hyper_ram_drv_TOP.vhd` pre komunikáciu s externou pamäťou DDR2 S27KS0641. Na nasledujúcej strane je jeho bloková schéma. Prevod medzi AXI-4 Interface a zbernicou HyperRam je realizovaný jadrom IP (Intellectual property) od spoločnosti Spansion `rpc2_ctrl_ip.v`. Pre zápis a čítanie z priestoru externej pamäte bolo potrebné vytvoriť dva stavové automaty, ktoré riadia transakcie na rozhraní tohto IP bloku cez zbernicu AXI. Bloková schéma ovládača je na nasledujúcej strane.



Obr. 39 Zapojenie ovládača externej DRAM pamäte S27KS0641

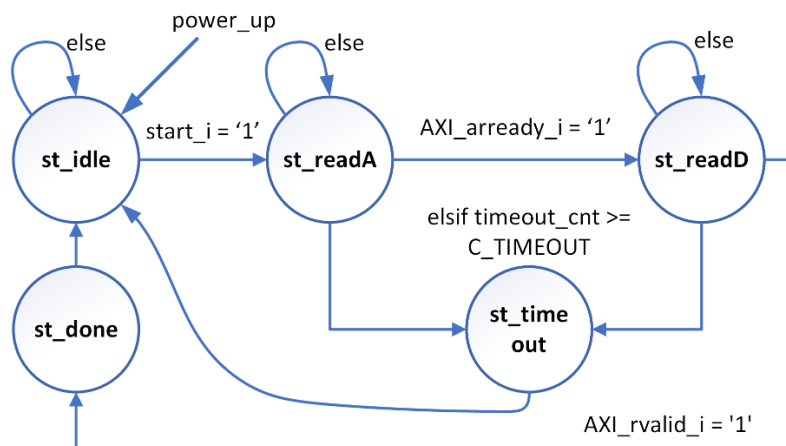
Riadiace automaty `AXIfsm_tx` a `AXIfsm_rx` sú v pozícii Mastera tzn. sú nadriadené nad IP blokom `rpc2_ctrl.v`, ktorý je v podriadenej pozícii. Zápis dát do podriadenej AXI rozhrania je inicializovaný signálom `start_i` kedy FSM prejde zo stavu `st_idle` do stavu pre zápis `st_writeA`. V tomto stave FSM asertuje signál `AXI_awvalid_o` a čaká na odpoveď informujúcej o pripravenosti pre zápis adresy signálom `AXI_awready_i`. V prípade, že odpoveď nepríde v rámci presne stanoveného časového limitu automat prechádza do stavu `st_timeout` a proces je uspatý. V stave `st_writeD` je asertovaný signál povolovací signál `AXI_wvalid_o` pre zápis dát do podriadenej AXI zariadenia. V tomto stave automat ostáva, pokiaľ nedostane odpoveď o pripravenosti zápisu do adresného priestoru signálom `AXI_wready_i`. Rovnako ako v `st_writeA` aj tu FSM v prípade neprijatia odpovede signálom `AXI_wready_i` v určitom časovom intervale prechádza do stavu `st_timeout`. FSM po úspešnom zápise prejde do stavu `st_done` kde vygeneruje

jednoplusný signál informujúci ďalšie vonkajšie bloky v SSPO o dokončení operácie zápisu.



Obr. 40 Diagram prechodov AXIfsm_tx.vhd

Čítanie dát z podriadeného AXI rozhrania je inicializované signálom start_i kedy FSM prejde zo stavu st_idle do stavu st_readA pre zápis adresy, z ktorej sa bude čítať. V tomto stave FSM asertuje signál AXI_arvalid_o a čaká na odpoveď informujúcej o pripravenosti pre zápis adresy signálom AXI_arready_i. V prípade, že odpoveď nepríde v rámci presne stanoveného časového limitu automat prechádza do stavu st_timeout a proces je uspatý. V stave st_readD je asertovaný signál povolovací signál AXI_rready_o informujúci o pripravenosti prijať dáta z podriadeného zariadenia AXI. V tomto stave automat stagnuje pokiaľ nedostane validné dáta. Rovnako ako v st_readA aj tu FSM v prípade neprijatia odpovede signálom AXI_arvalid_i v určitom časovom intervale prechádza do stavu st_timeout. Po úspešnom prečítaní jedného slova FSM prejde do stavu st_done kde vygeneruje jednoplusný signál informujúci ďalšie vonkajšie bloky v SSPO o dokončení operácie čítania. [18]



Obr. 41 Diagram prechodov AXIfsm_rx.vhd

Tab. 8 Štatistika využitia zdrojov v FPGA po syntéze

	Využitých [-]	Celkový počet dostupných [-]	Percentuálne využitie [%]
FF	2504	41600	6,02
LUT	1772	20800	8,52
LUTRAM	160	9600	1,67
BRAM	4	50	8,00

Pre úspešné implementovanie procesu extrakcie statického pozadia resp. získanie rozdielových snímok z teoretickej časti tejto práce je v budúcnosti potrebné využiť ďalší nadriadený stavový automat, ktorý bude riadiť vyššie uvedené stavové automaty. Principiálna realizácia je jednoduchá a je veľmi podobná s implementáciou v testovacej aplikácii uvedenej v kapitole 9.6. Proces bude zahájený vložením dvojice snímok do pamäte S27KS0641. Celkovo bude tak v pamäti uložených 960 kB obrazových informácií. Po úspešnom zápise dvojice snímok bude vytvorená rozdielová snímka, ktorá bude taktiež uložená v pamäti S27KS0641. Následne budú z pamäte vyčítavané okienkové bloky o veľkosti 8x8px, ktoré budú prekopírované do internej blokovej pamäte v FPGA. V týchto okienkach o veľkosti 64 bajtov sa pomocou akumulátora a jednoduchého bitového posunu vypočíta priemerný pohyb, ďalej označovaný ako konštanta THR. Numerická hodnota priemerného pohybu bude komparovaná s jednotlivými diferenciami v okienkovom priestore. V prípade ak, bude diferencia menšia ako THR bude tento bod považovaný za statický. Naopak, ak bude hodnota diferencie väčšia, bude na tejto pozícii v novej blokovej pamäti vytvorený binárny bod s hodnotou log.1. [19]

ZÁVER

Cieľom tejto práce bolo preskúmať, navrhnúť a otestovať riešenie systému pre sledovanie pohybujúcich sa objektov SSPO. Prvá časť práce je venovaná skúmaniu jednotlivých postupov nevyhnutných pre vytvorenie sledovacieho zariadenia, ktoré sú súčasťou dosiahnutých výsledkov diplomovej práce. Druhá časť sa zaoberá praktickou realizáciou SSPO.

Pri návrhu mechanickej časti SSPO bol kladený dôraz na jej vyrobiteľnosť a kompatibilitu s voľne dostupnými kamerami a statívmi. Ďalej bol vykonaný prieskum trhu, kde bol zistený silný deficit cenovo dostupných produktov, ktoré by boli schopné konkurovať SSPO.

Následne bola preskúmaná metodika získania pohybujúcich sa objektov v obraze s predložením riešenia rozdielu troch snímok, ktoré je ideálne pre implementáciu do obvodov FPGA. Dôležitou časťou práce bol návrh tejto metódy na úrovni dátových tokov s následnou filtráciou šumu. Po získaní zmeny polohy objektu v obrazovom snímku bol skoncipovaný teoretický výpočet súradníc sledovaného objektu. Ďalej bolo v testovacej aplikácii (ktorá je dostupná v informačnom systéme) zistené, že ak sa sledovaný objekt pohybuje príliš pomaly, nebude možné jeho zachytenie. Vyplýva to z princípu sledovania uvedeného v kapitole 6. Tento problém je možné odstrániť dvomi paralelne bežiacimi procesmi, kedy jeden proces pracuje priamo s frekvenciou kamery a druhý vynecháva niektoré snímky a odčítava medzi sebou napríklad každý desiaty.

V praktickej časti realizácie SSPO bola vyrobená mechanicko-motorická časť s použitím vývojových prostriedkov CAD SolidWorks. Na overenie a odsimulovanie teoretických predpokladov tejto práce bola vytvorená univerzálna testovacia platforma bližšie popísaná v kapitole 9.6. Predmetom tejto práce nie je podrobný popis tejto testovacej platformy ale následné riešenie implementácie v obvode FPGA, ktoré presahuje rozsah diplomovej práce. Je nutné podotknúť, že všetky teoretické predpoklady vedúce k úspešnej implementácii systému do obvodu FPGA sa preukázali ako realizovateľné.

Po úspešnom doladení sledovacích algoritmov v testovacej aplikácii bol vybraný vhodný finálny realizačný mikromodul TE0725 postavený na báze obvodu FPGA Artix-7. Na tento čip bol vytvorený koncept riadiacej jednotky, ktorej úlohou je vykonávanie všetkých funkcií uvedených v teoretickej časti tejto práce. Ďalej bol navrhnutý a implementovaný dekodér vstupných dát z kamery OV 2640 na paralelnú kombináciu RGB, prevodník týchto farebných zložiek na zložku luminiscenčnú. Tieto komponenty boli odsimulované a otestované pomocou ILA (Integrated Logical Analyser), ktorý je súčasťou vývojového prostredia Xilinx Vivado. Ďalej bolo implementované zjednodušené rozhranie I²C pre konfiguráciu kamery OV 2640.

Je na mieste poukázať na to, že dokumentácia tejto kamery od spoločnosti OmniVision je nedostačujúca a návrhár musí veľa parametrov „domýšľať“ čo zaberie nesmierne veľa času. Následne po nastavení kamery bol vyvinutý a implementovaný verifikačný kanál z FPGA do aplikácie v PC. Jeho účelom je prenášanie kamerového obrazu a ďalších nevyhnutných dát ako sú napríklad rozdielové snímky alebo poloha sledovaného objektu v obraze.

Ďalej boli navrhnuté a behaviorálne otestované periférie určené pre mechanický pohyb kamery, ako sú ovládač krokového motora a serva. Z časových dôvodov a problémov, ktoré pri komunikácii s kamerou nastávali, nebolo jadro systému z kapitoly 9.5 na obr. 16 plne implementovaná a preto nebolo možné dokončiť otestovanie tohto systému v reálnom prostredí. Aj napriek tomuto faktoru je veľmi pravdepodobné, že po úspešnom doladení bude systém vykazovať lepšie parametre ako testovacia platforma s webkamerou a PC aplikáciou.

Treba podotknúť, že implementácia funkčného designu do FPGA nie je iba o jeho návrhu s úspešnou syntézou ale aj o definovaní obmedzujúcich podmienok. Typickými príkladmi sú napríklad maximálne prípustné časové oneskorenia na dátových linkách medzi internou logikou FPGA a externým čipom, ďalej definície asynchrónnych hodinových domén, presunutie klopných obvodov do vstupno/výstupných bánk danej architektúry a mnoho ďalších. Všetky zdrojové súbory, vytvorené projekty a výrobné dáta sú priložené v elektronickej podobe v informačnom systéme.

Overenie funkčnosti systému v reálnom prostredí mojej pracovnej dielne na dočasnej testovacej platforme zloženej z mikro-kontroléru, pomocných obvodov a Windows aplikácie si môže čitateľ pozrieť na mojom youtubovom kanáli pod linkom: https://www.youtube.com/channel/UCb_3Ew9e0QC3nVyuTqmLE_w?view_as=subscriber

POUŽITÁ LITERATÚRA

- [1] SOLOSHOT3. *Soloshot* [online]. USA: soloshot, 2018 [cit. 2018-12-13]. Dostupné z: <https://soloshot.com/collections/soloshot3>
- [2] PIXIO: *Your Personal Auto-Follow Camera for INDOOR & OUTDOOR Activities* [online]. USA: Movensee, 2018 [cit. 2018-12-13]. Dostupné z: <https://shop.movensee.com/en/pixio-and-pixem-robot-cameraman/18-pixio-robot-cameraman.html>
- [3] RGB color model. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-12-13]. Dostupné z: https://en.wikipedia.org/wiki/RGB_color_model
- [4] Bayer filter. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-12-13]. Dostupné z: https://en.wikipedia.org/wiki/Bayer_filter
- [5] YCbCr. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-12-13]. Dostupné z: <https://en.wikipedia.org/wiki/YCbCr>
- [6] *Hardware Design of Moving Object Detection on Reconfigurable System* [online]. Taiwan: Hung-Yu Chen, Yuan-Kai Wang, 2016 [cit. 2018-12-14]. Dostupné z: https://file.scirp.org/pdf/JCC_2016081015273428.pdf
- [7] Foreground detection. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-12-14]. Dostupné z: https://en.wikipedia.org/wiki/Foreground_detection
- [8] Šum. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-12-13]. Dostupné z: <https://cs.wikipedia.org/wiki/%C5%A0um>
- [9] *ŠUM V DIGITÁLNÍ FOTOGRAFII* [online]. Praha: Jaroslav Fikker, 2004 [cit. 2018-12-13]. Dostupné z: <https://www.fotoaparar.cz/clanek/323/sum-v-digitalni-fotografii-7193/>
- [10] *Pololu - Stepper Motor: Bipolar* [online]. Pololu Corporation 920 Pilot Rd. Las Vegas, NV 89119 USA: Pololu, 2012 [cit. 2018-12-13]. Dostupné z: <https://www.pololu.com/file/0J690/SY35ST26-0284A.pdf>

- [11] *OV2640_DS (1.6).fm* [online]. Sunnyvale, CA USA: OmniVision, 2006 [cit. 2018-12-13]. Dostupné z: https://www.waveshare.com/w/upload/6/6b/OV2640_DS%281.6%29.pdf
- [12] *OV2640 Camera Board module, 2 Megapixel* [online]. USA: WAVESHARE, 2017 [cit. 2018-12-13]. Dostupné z: <https://www.waveshare.com/ov2640-camera-board.htm>
- [13] *Extending 28nm Leadership* [online]. San Jose, California, U.S: Xilinx, 2011 [cit. 2018-12-13]. Dostupné z: <https://www.xilinx.com/products/silicon-devices/28nm-extensions.html>
- [14] *TE0725 TRM - Public Docs - Trenz Electronic Wiki* [online]. DE: trenz electronic, 2018 [cit. 2018-12-13]. Dostupné z: <https://wiki.trenz-electronic.de/display/PD/TE0725+TRM>
- [15] *DC/DC Converters: TSR 0.5 Series, 0.5 A Switching Regulator. Tracopower* [online]. Baar, 2016 [cit. 2016-12-12]. Dostupné z: <http://www.tracopower.com/products/tsr05.pdf>
- [16] *LD1117A* [online]. USA: ST, 2013 [cit. 2019-05-20]. Dostupné z: <https://www.st.com/resource/en/datasheet/cd00002116.pdf>
- [17] *PMEG3010ER: 1 A low VF MEGA Schottky barrier rectifier* [online]. USA: NXP, 2017 [cit. 2019-05-20]. Dostupné z: <https://assets.nexperia.com/documents/data-sheet/PMEG3010ER.pdf>
- [18] *AMBA® AXI™ and ACE™ Protocol Specification: AXI3™, AXI4™, and AXI4-Lite™ ACE and ACE-Lite™* [online]. USA: ARM, 2011 [cit. 2019-05-20]. Dostupné z: http://www.gstitt.ece.ufl.edu/courses/fall15/eel4720_5721/labs/refs/AXI4_specification.pdf
- [19] *S27KL0641/S27KS0641: HyperRAM™ Self-Refresh DRAM* [online]. USA: CYPRESS, 2018 [cit. 2019-05-20]. Dostupné z: <https://www.cypress.com/file/183506/download>

ZOZNAM SYMBOLOV VELIČÍN A SKRATIEK

2D	Two – Dimensional
3D	Three – Dimensional
A	Ampér
ABS	Akrylonitrilbutadiénstyrén
AEG	Automatic Exposure Gain
ARGB	Alpha Red Green Blue
AV	Analog Video
B	Blue
BGA	Ball Grid Array
C	Center
CAD	Computer Aided Design
Cb	Blue Chrominance
CLK	Clock
CMOS	Complementary Metal Oxid Semiconductor
Cr	Red Chrominance
DPS	Doska Plošných Spojov
DRAM	Dynamic Random Access Memory
DSP	Digital Signal Processor
DVP	Digital Video Port
EEPROM	Electrically Erasable Programmable Read-Only Memory
FIFO	First in First out
FOV	Field of View
FPGA	Field Programmable Gate Array

FSM	Finite State Machine
GPS	Global Positioning System
I ² C	Inter-Integrated Circuit
JTAG	Joint Test Action Group
Li-Pol	Lítium-Polymér
R	Red
RFID	Radio Frequency Identification
RGB	Red Green Blue
RJ	Riadiaca Jednotka
RTL	Register Transfer Logic
SCCB	Serial Camera Control Bus
SoC	System on Chip
SSPO	Systém pre Sledovanie Pohybujúcich sa Objektov
SVGA	Super Video Graphic Array
USB	Universal Serial Bus
UXGA	Ultra Extended Graphic Array
VHSIC	Very High Speed Integrated Circuit
VHDL	VHSIC Hardware Description Language

ZOZNAM OBRÁZKOV

Obr. 1 Grafické znázornenie rozloženia SSPO v priestore.....	1
Obr. 2 Mechanické usporiadanie systému pre sledovanie pohybujúcich sa objektov	2
Obr. 3 Soloshot 65 a PIXIO Zdroje: https://shop.movensee.com/165-large_default/pixio-robotcameraman.jpg , PIXIO-robot-cameraman.jpg	3
Obr. 4 Koncept systému SSPO	4
Obr. 5 Grafické znázornenie relácie zorného poľa kamery s pohybujúcim sa cieľom.....	6
Obr. 6 Fyzická štruktúra obrazového snímača a jeho výstupný dátový formát [4]	8
Obr. 7 Znázornenie procesu extrakcie diferenčného snímku dif_{pos}	10
Obr. 8 Princíp okienkovej filtrácie	13
Obr. 9 Grafické znázornenia okienkovej filtrácie vstupného zašumeného snímku dif_{nN} na výsledný dif_{ndN} snímok vo formáte jednobitového dvojrozmerného poľa.....	13
Obr. 10 Znázornenie výpočtu približného stredy sledovaného objektu	15
Obr. 11 Znázornenie vzdialenosti sledovaného objektu od stredy obrazu v pixeloch ...	16
Obr. 12 Pohľad na 3D model obalu krokového motora vo vývojovom prostredí CAD SolidWorks2018	17
Obr. 13 Kamerový modul WAVESHARE 8532 s čipom OV 2640 Zdroj: https://www.waveshare.com/.../ov2640-camera-board_1_1_5.jpg	18
Obr. 14 Blokový diagram modulu TE0275-3 Zdroj: https://wiki.trenz-electronic.de/download/	19
Obr. 15 Blokové znázornenie RJ v systéme SSPO.....	20
Obr. 16 Koncept RJ v hradlovom poli FPGA Artix-7	21
Obr. 17 Priebeh synchronizačných signálov VSYNC, HREF vzhľadom k dátam Y Zdroj: https://www.uctronics.com/download/cam_module/OV2640DS.pdf	22
Obr. 18 Výstupný formát dát UXGA RGB RAW z kamery OV2640 Zdroj: https://www.uctronics.com/download/cam_module/OV2640DS.pdf	23
Obr. 19 Zjednodušené blokové zapojenie DVP dekodéra <i>DVP_decoder.vhd</i>	24
Obr. 20 Definícia prechodov medzi stavmi FSM <i>Control_FSM.vhd</i>	27
Obr. 21 Prevodník z RGB na Y (<i>rgb_to_y.vhd</i>)	28
Obr. 22 Výsledný syntetizovaný návrh <i>rgb_to_y.vhd</i> na úrovni RTL.....	29
Obr. 23 Grafické znázornenie testovacieho systému.....	30
Obr. 24 Schematické zapojenie vinutia krokového motora do H-mostíka Zdroj: https://www.st.com/content/ccc/resource/technical/document/application_note/57/c8/7c/c1/0d/91/46/89/CD00003774.pdf/files/CD00003774.pdf/jcr:content/translations/en.CD00003774.pdf	31
Obr. 25 Principiálny diagram bipolárneho pol krokového budenia krokového motora Zdroj: https://www.st.com/content/ccc/resource/technical/document/application_note/57/c8/7c/	

c1/0d/91/46/89/CD00003774.pdf/files/CD00003774.pdf/jcr:content/translations/en.CD00003774.pdf	32
Obr. 26 Zobrazenie podmienených prechodov FSM s výstupom Phase	33
Obr. 27 Bloková schéma distribúcie napájania na RJ	34
Obr. 28 Obr. Schematické zapojenie distribúcie napájacích napätí na DPS RJ	36
Obr. 29 Demonštrácia funkčnosti lokalizácie pohybujúceho sa objektu v statickom obraze z videosekvencie	37
Obr. 30 Bloková schéma <i>filtration_unit.vhd</i>	39
Obr. 31 Približné schematické začlenenie jednotky filtrácie <i>filtration_unit.vhd</i> do celého systému	40
Obr. 32 Stavový diagram prechodov Mealyho stavového automatu	41
Obr. 33 Blokové znázornenie testovacieho jednotky <i>filtration_unit.vhd</i>	43
Obr. 34 Porovnanie vstupného zašumeného obrázka <i>in_image.bmp</i> (nad červenou čiarou) s výsledným odfiltrovaným obrázkom <i>result_image.bmp</i>	43
Obr. 35 Zapojenie najnižšej komunikačnej vrstvy (ovládača) medzi FT232 a SSPO....	45
Obr. 36 Stavový diagram Mealyho vysielacieho FSM pre FT232H	46
Obr. 37 Štruktúra paketu	48
Obr. 38 Diagram prechodov generátora ESP znakov	49
Obr. 39 Zapojenie ovládača externej DRAM pamäte S27KS0641	50
Obr. 40 Diagram prechodov <i>AXIfsm_tx.vhd</i>	51
Obr. 41 Diagram prechodov <i>AXIfsm_rx.vhd</i>	51

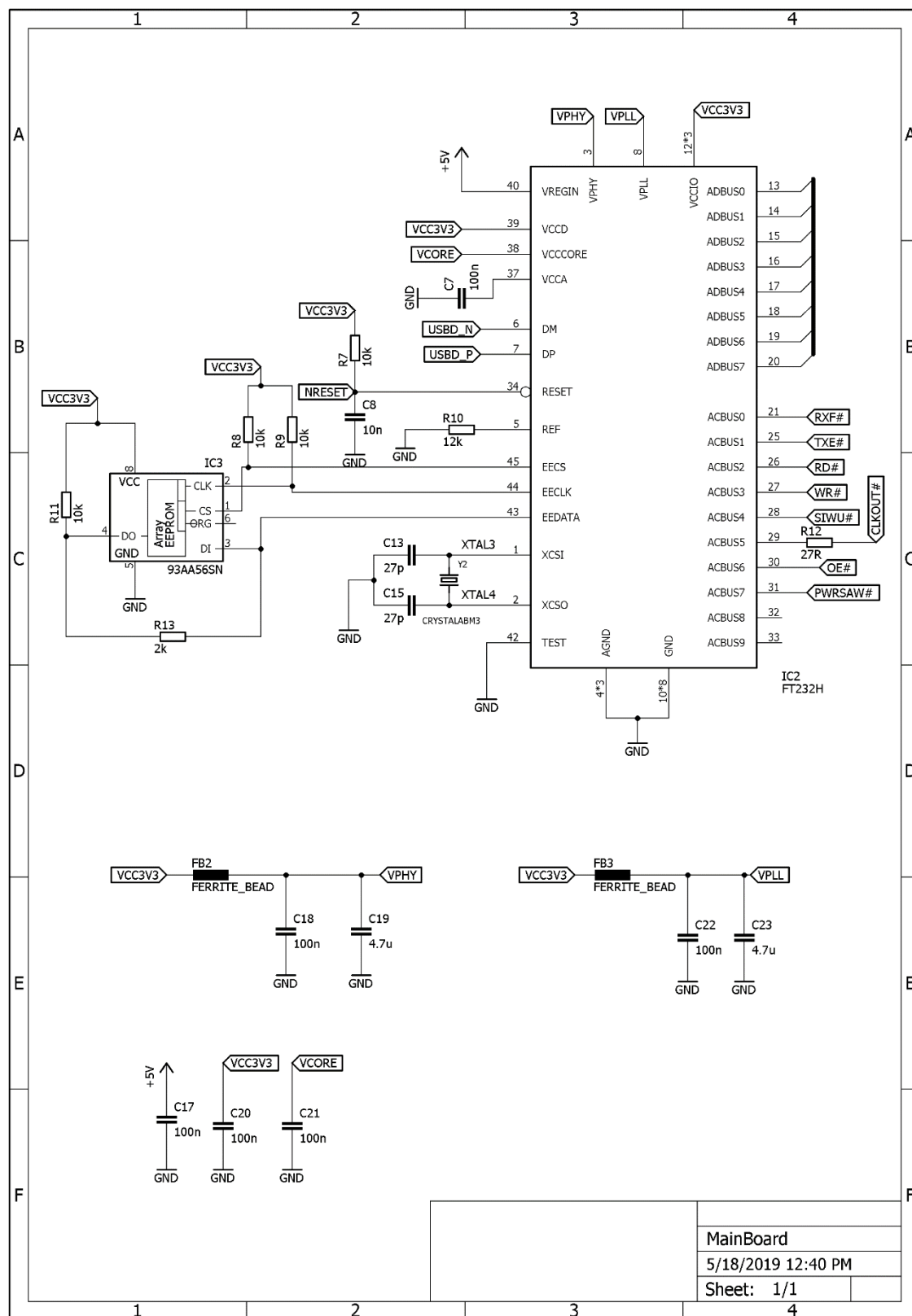
ZOZNAM TABULIEK

Tab. 1 Prehľad základných parametrov komerčne dostupných výrobkov	3
Tab. 2 Hodnoty závislých výstupov <i>Control_FSM.vhd</i> na aktuálnom stave a vstupe <i>data_valid_i</i>	25
Tab. 3 Popis prúdového odberu a spôsob napájania jednotlivých zariadení	34
Tab. 4 Parametre DC/DC konvertora Traco Power TSR 0.5-2433	35
Tab. 5 Štatistika využitia zdrojov v FPGA po syntéze	44
Tab. 6 Štatistika využitia zdrojov v FPGA po syntéze	47
Tab. 7 Štatistika využitia zdrojov v FPGA po syntéze	49
Tab. 8 Štatistika využitia zdrojov v FPGA po syntéze	52

A.1 Zapojenie konektorov s distribúciou napájania



A.2 Zapojenie obvodu FT232H s konfiguračnou pamäťou

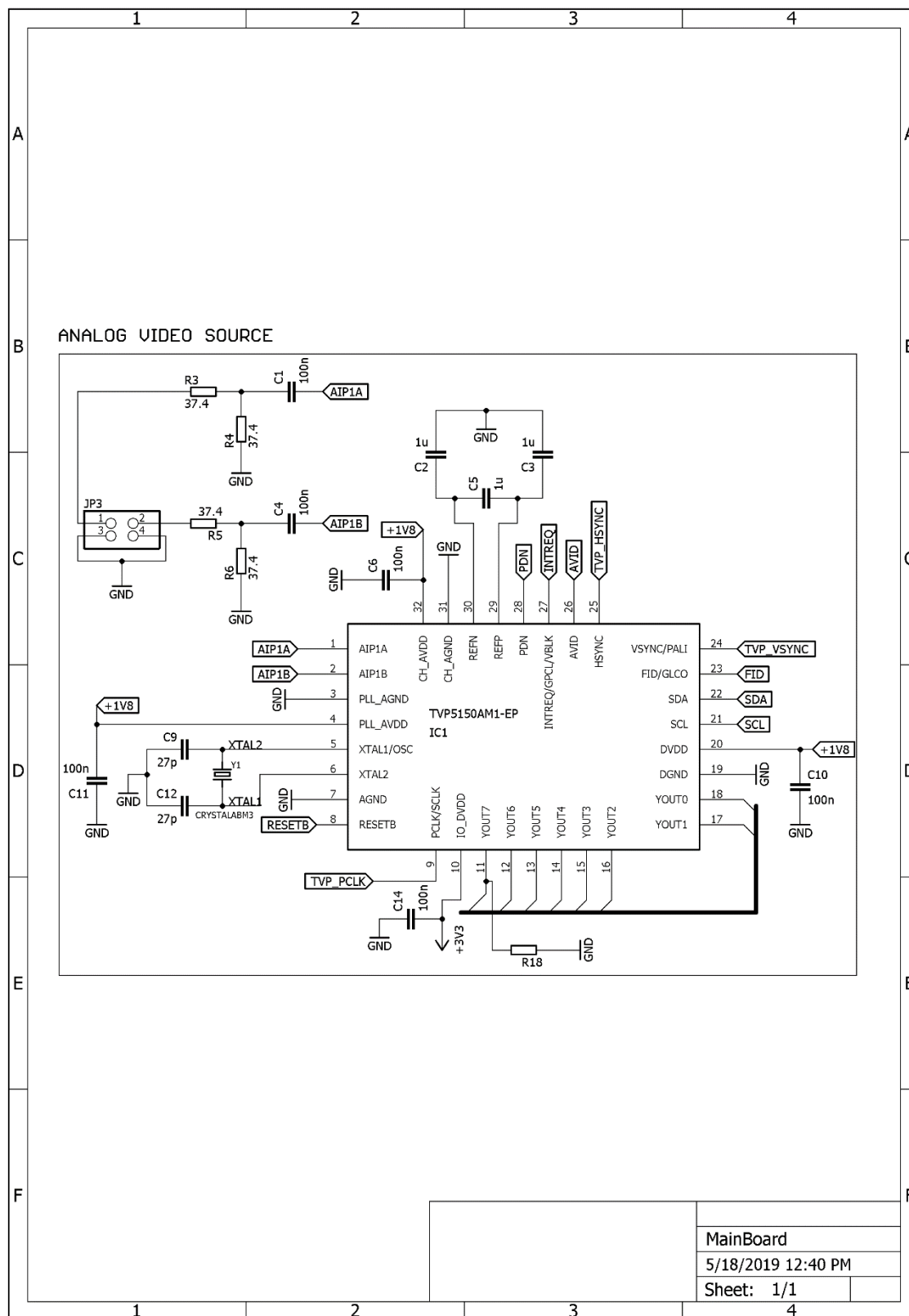


The schematic diagram illustrates the MainBoard's pin headers and internal components. It is organized into a grid with columns 1 through 4 and rows A through F.

- Pin Headers:**
 - JB1:** A 48-pin header with pins 1-48. Pins 1-21 are labeled with functions like SCL, OV_HREF, OV_XCLK, OV_Y8, OV_Y6, OV_Y4, OV_Y2, OV_PWDN, TVP_HSYNC, FID, TVP_YOUT1, TVP_YOUT3, TVP_YOUT5, TVP_YOUT7, AVID, PDN, GPIO0, PH_B1_I, PH_A1_I, and SERVO_PWM0. Pins 22-48 are labeled with functions like GND, B35_L23_N, B35_L15_P, B35_L13_N, B35_L13_P, B35_L12_N, B35_L12_P, B35_L22_N, B35_L22_P, B35_L17_N, B35_L17_P, B35_L18_N, B35_L18_P, B35_L14_N, B35_L14_P, B35_L16_P, B35_L16_N, B35_L19_N, B35_L19_P, B35_L10_P, B35_L10_N, B35_L8_N, B35_L8_P, B35_L11_N, B35_L11_P, B35_L3_N, B35_L3_P, B35_L2_N, B35_L2_P, B35_L7_N, B35_L7_P, B35_L1_N, B35_L1_P, B35_L5_N, B35_L5_P, B35_L6_N, B35_L6_P, B35_L19_N, B35_L19_P, and SERVO_PWM1.
 - JB2:** A 98-pin header with pins 53-98. Pins 53-74 are labeled with functions like B34_L24_N, B34_L24_P, B34_L21_N, B34_L21_P, B34_L18_N, B34_L18_P, B34_L22_N, B34_L22_P, B34_L20_N, B34_L20_P, B34_L23_N, B34_L23_P, B34_L10_N, B34_L10_P, B34_L19_P, B34_L19_N, B34_L8_P, B34_L8_N, B34_L9_N, B34_L9_P, B34_L7_N, B34_L7_P, B34_L13_P, B34_L13_N, B34_L12_P, B34_L12_N, B34_L11_N, B34_L11_P, B34_L14_P, B34_L14_N, B34_L16_N, B34_L16_P, B34_L17_N, B34_L17_P, B34_L15_N, B34_L15_P, B34_L3_N, B34_L3_P, B34_L1_N, B34_L1_P, and B34_L4_P, B34_L4_N. Pins 75-98 are labeled with functions like NRESET, OE#, SIWU#, RD#, RXF#, ADBUS6, ADBUS4, ADBUS2, ADBUS0, and ADBUS1.
 - FT232 REGION:** A 93-pin header with pins 53-93. Pins 53-74 are labeled with functions like B34_L24_N, B34_L24_P, B34_L21_N, B34_L21_P, B34_L18_N, B34_L18_P, B34_L22_N, B34_L22_P, B34_L20_N, B34_L20_P, B34_L23_N, B34_L23_P, B34_L10_N, B34_L10_P, B34_L19_P, B34_L19_N, B34_L8_P, B34_L8_N, B34_L9_N, B34_L9_P, B34_L7_N, B34_L7_P, B34_L13_P, B34_L13_N, B34_L12_P, B34_L12_N, B34_L11_N, B34_L11_P, B34_L14_P, B34_L14_N, B34_L16_N, B34_L16_P, B34_L17_N, B34_L17_P, B34_L15_N, B34_L15_P, B34_L3_N, B34_L3_P, B34_L1_N, B34_L1_P, and B34_L4_P, B34_L4_N. Pins 75-93 are labeled with functions like NRESET, OE#, SIWU#, RD#, RXF#, ADBUS6, ADBUS4, ADBUS2, ADBUS0, and ADBUS1.
- Internal Components:**
 - JP4:** A 4-pin header with pins 1-4. Pins 1-2 are labeled with functions like GPIO0 and GPIO1. Pins 3-4 are labeled with functions like PH_B1_I and PH_A1_I.
 - JP5:** A 4-pin header with pins 1-4. Pins 1-2 are labeled with functions like PH_A1 and PH_B1. Pins 3-4 are labeled with functions like PH_A0 and PH_B0.
 - IC5:** An LV8548MCSOIC-10 IC with pins 1-10. Pins 1-4 are labeled with functions like VCC, IN1, IN2, IN3, IN4, and GND. Pins 5-10 are labeled with functions like OUT1, OUT2, OUT3, OUT4, and GND.

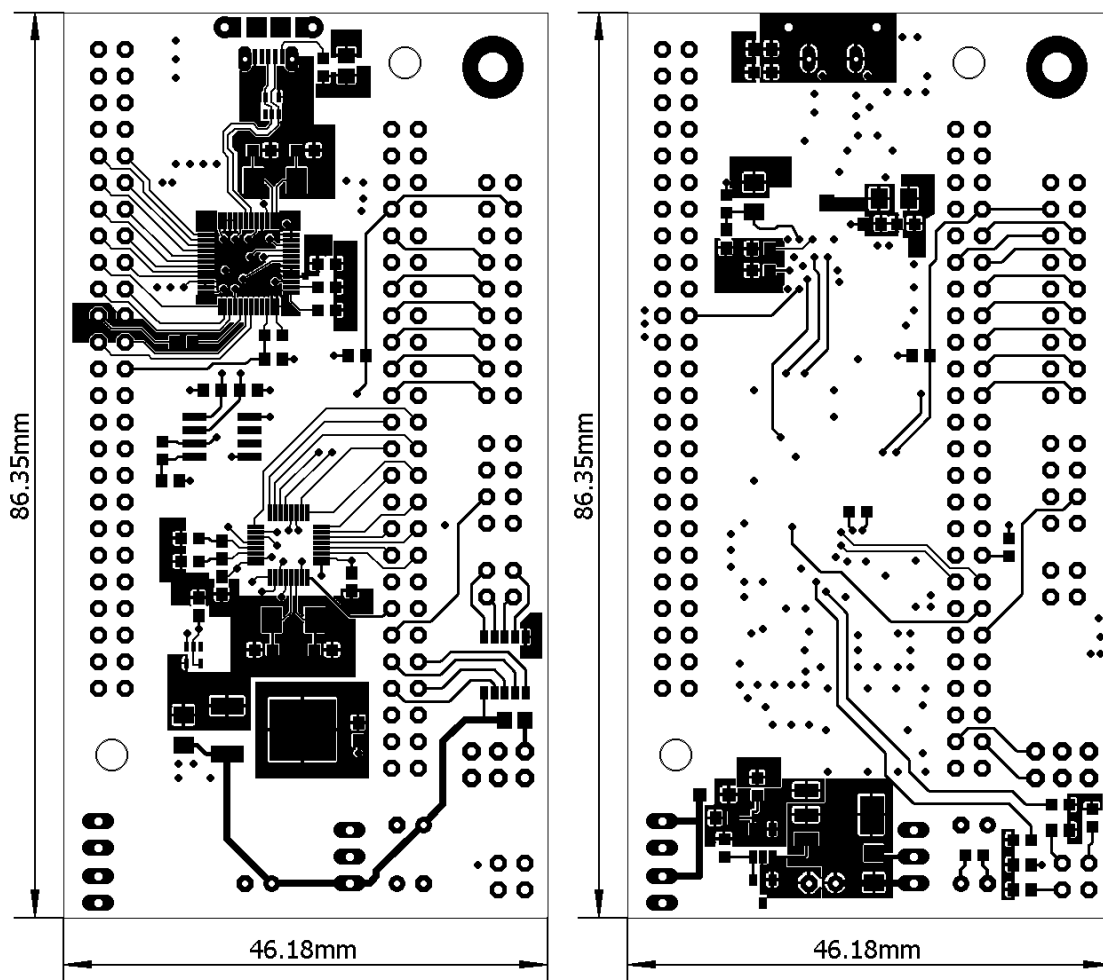
The diagram also shows power supply connections (+3V3, GND) and a capacitor (C25) connected to the VCC pin of IC5.

A.4 Zapojenie rezervného zdroja video signálu

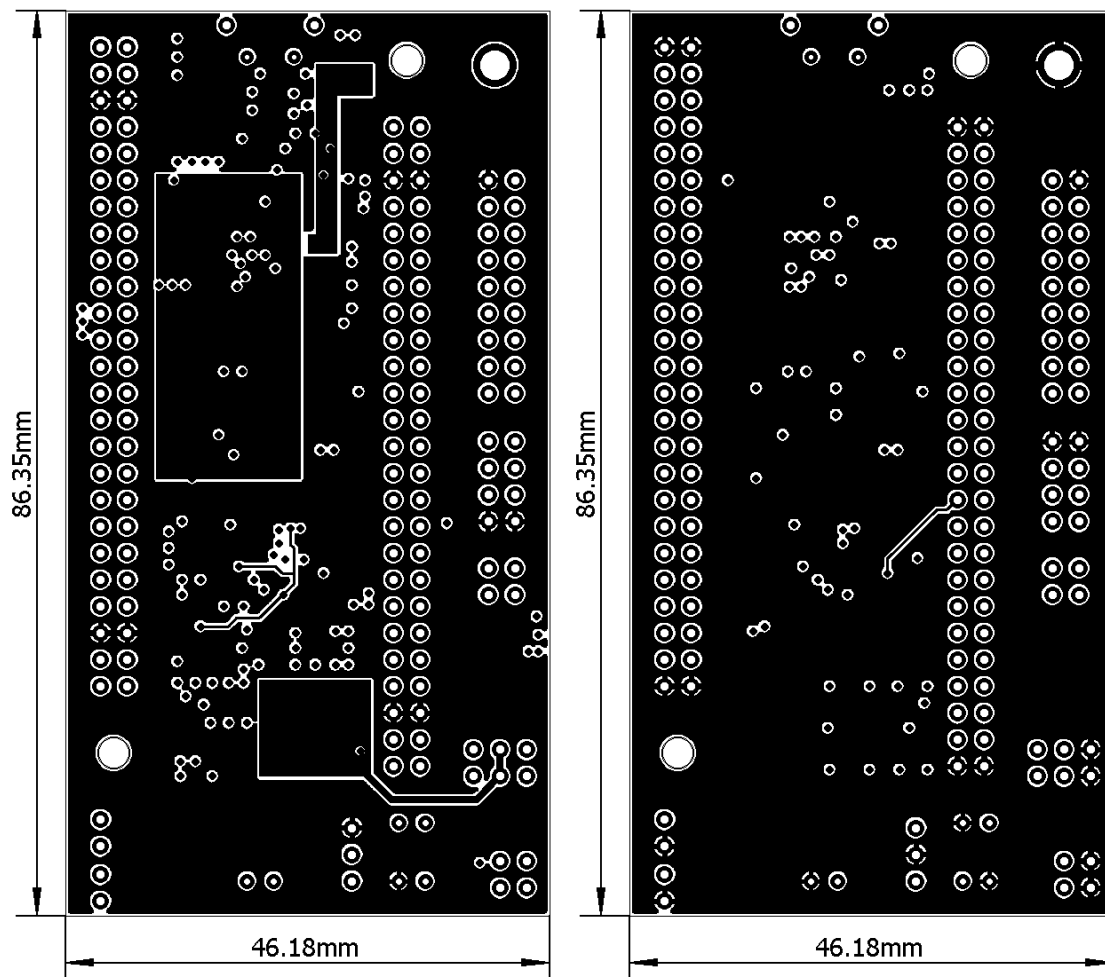


B DOSKA PLOŠNÝCH SPOJOV

B.1 Vodivé vrstvy - TOP (vľavo) a BOT (vpravo)



B.2 Vnútoraná zemniaca vrstva č. 2 (vpravo) a vnútoraná napájacia vrstva č. 15 (vľavo)



B.3 DPS osadenie súčiastok – horná vrstva (vľavo), spodná vrstva (vpravo)

